ADVANCED CRYPTOGRAPHIC PRIMITIVES

PART 2: CONSTRUCTIONS OF IBE / ABE

ACM INDIA SUMMER SCHOOL (12 - 06 - 2024)

Venkata Koppula (IIT Delhi)
kvenkata@iitd.ac.in

PLAN FOR THE TALK

n

Lattice based constructions of IBE and ABE

1. *Lattice Toolkit*
   - *1a. Learning with Errors Problem*
   - *1b. Lattice trapdoors*
2. *IBE constructions*
   - *2a. IBE scheme secure in the random oracle model*
   - *2b. IBE scheme secure in the standard model*
3. *ABE constructions (inner product, general circuits)*

TOOLKIT FOR
LATTICE BASED CRYPTOGRAPHY

NOTATIONS

All computation is mod q, where q is a large modulus

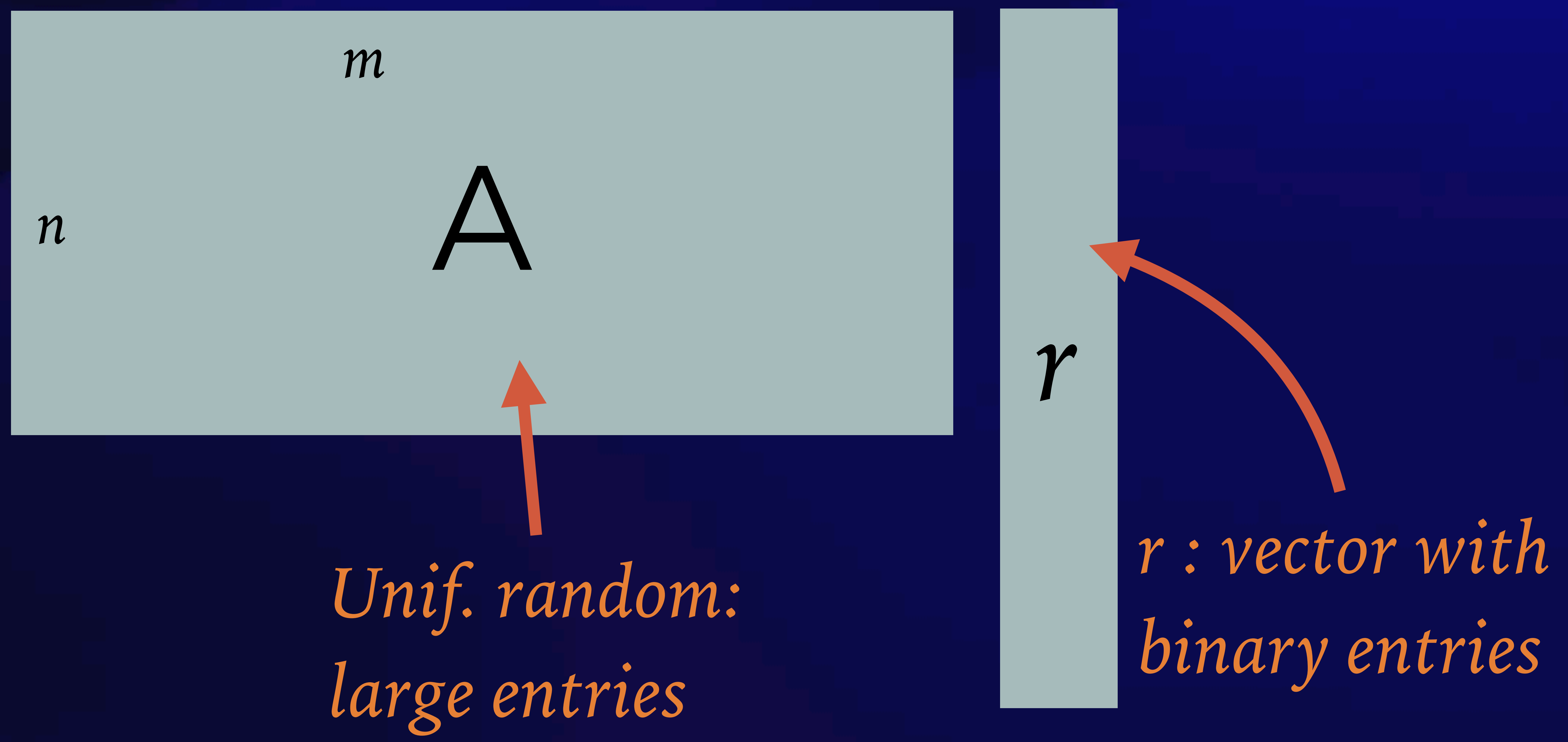
n

n : *security parameter*, $m \approx n^2$, $q \approx 2^{\sqrt{n}}$

small entries : poly(n) *large entries* : superpoly(n)

Example 1:

$\mathbf{A} \cdot \mathbf{r}$ *has large entries*



Unif. random: large entries

r : vector with binary entries

NOTATIONS

All computation is mod q, where q is a large modulus

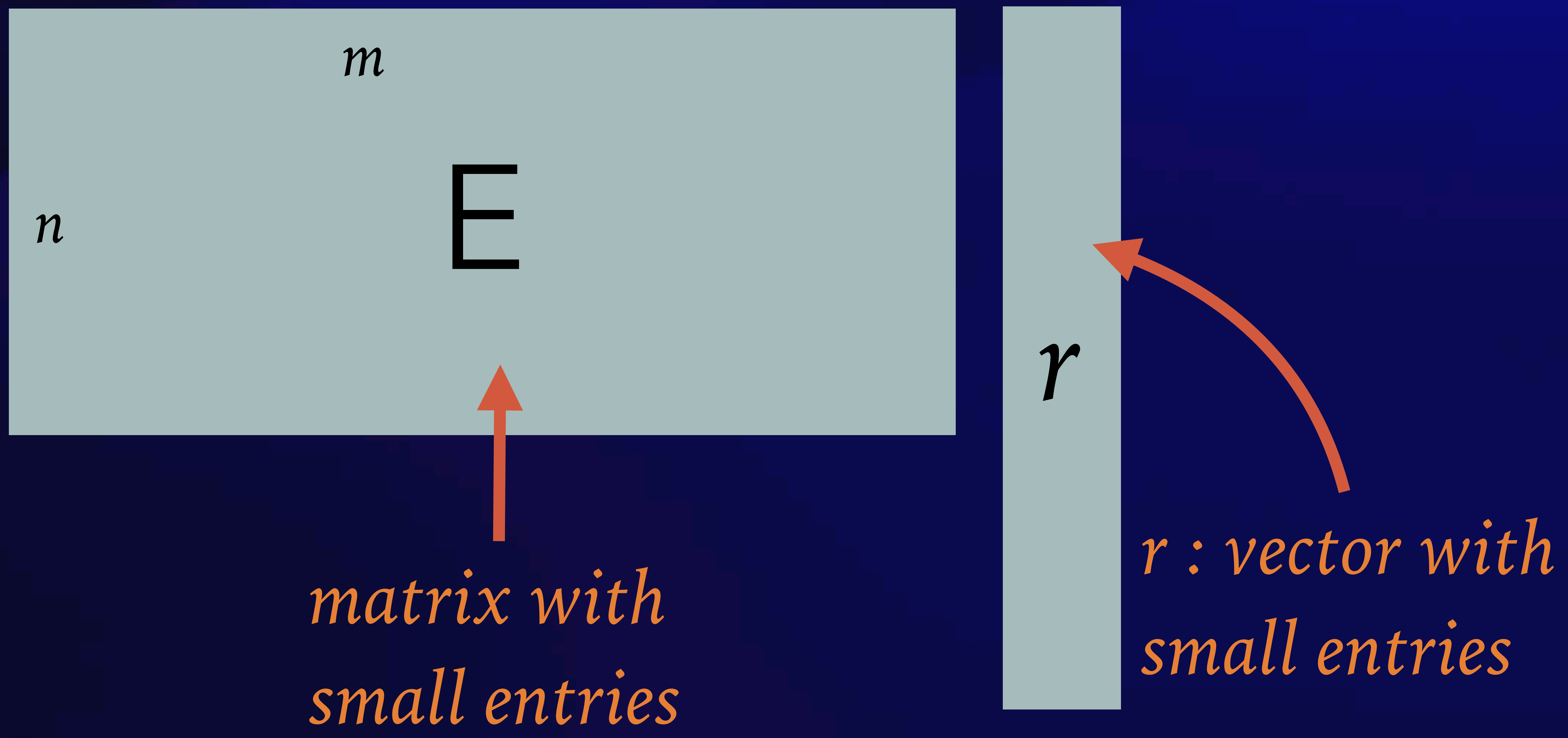
n

n : *security parameter*, $\quad m \approx n^2, \quad q \approx 2^{\sqrt{n}}$

small entries : $\mathsf{poly}(n)$ *large entries* : $\mathsf{superpoly}(n)$

Example 2:

$\mathbf{E} \cdot \mathbf{r}$ *has small entries*



matrix with small entries

r : vector with small entries

A USEFUL LEMMA

$m > n \log q$
n
A
A
r
$\approx$
A
u

r : vector with binary entries

(All computation is mod q, where q is a large modulus)

THE LEARNING WITH ERRORS PROBLEM

A

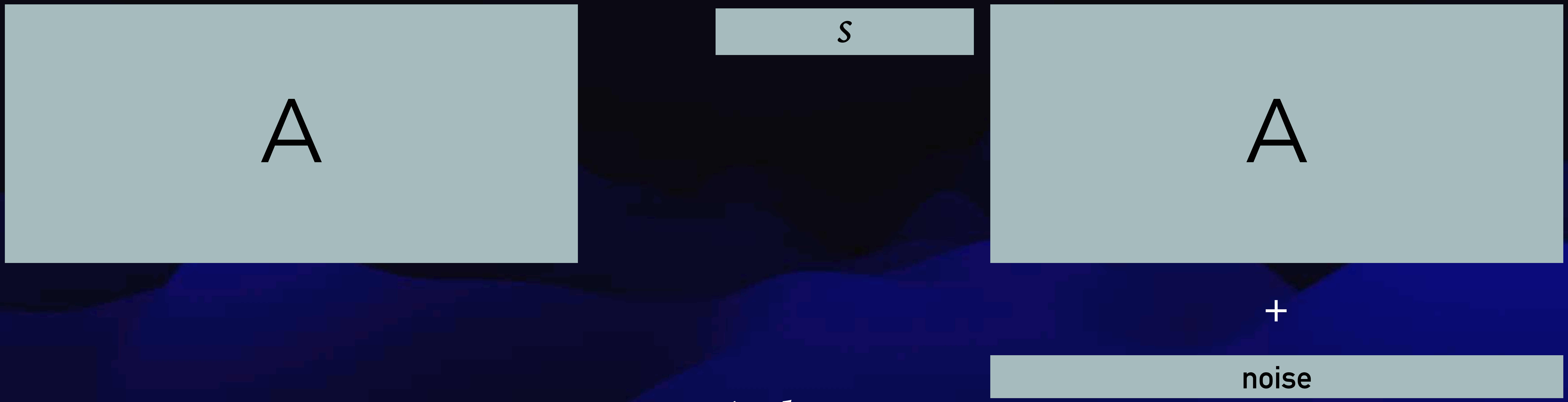
s

A

Find s.

(All computation is mod q, where q is a large modulus)

THE LEARNING WITH ERRORS PROBLEM



A

s

A

+

noise

Find s.

(All computation is mod q, where q is a large modulus)

THE LEARNING WITH ERRORS PROBLEM

$$\left\{ \left(\mathbf{A} \quad , \quad \mathbf{s}^{\mathrm{T}} \cdot \mathbf{A} + \mathbf{e} \right) \ : \ \mathbf{A} \leftarrow \mathbb{Z}^{n \times m}, \mathbf{s} \leftarrow \mathbb{Z}_q^n, \mathbf{e} \leftarrow [-B, B]^m \right\}$$

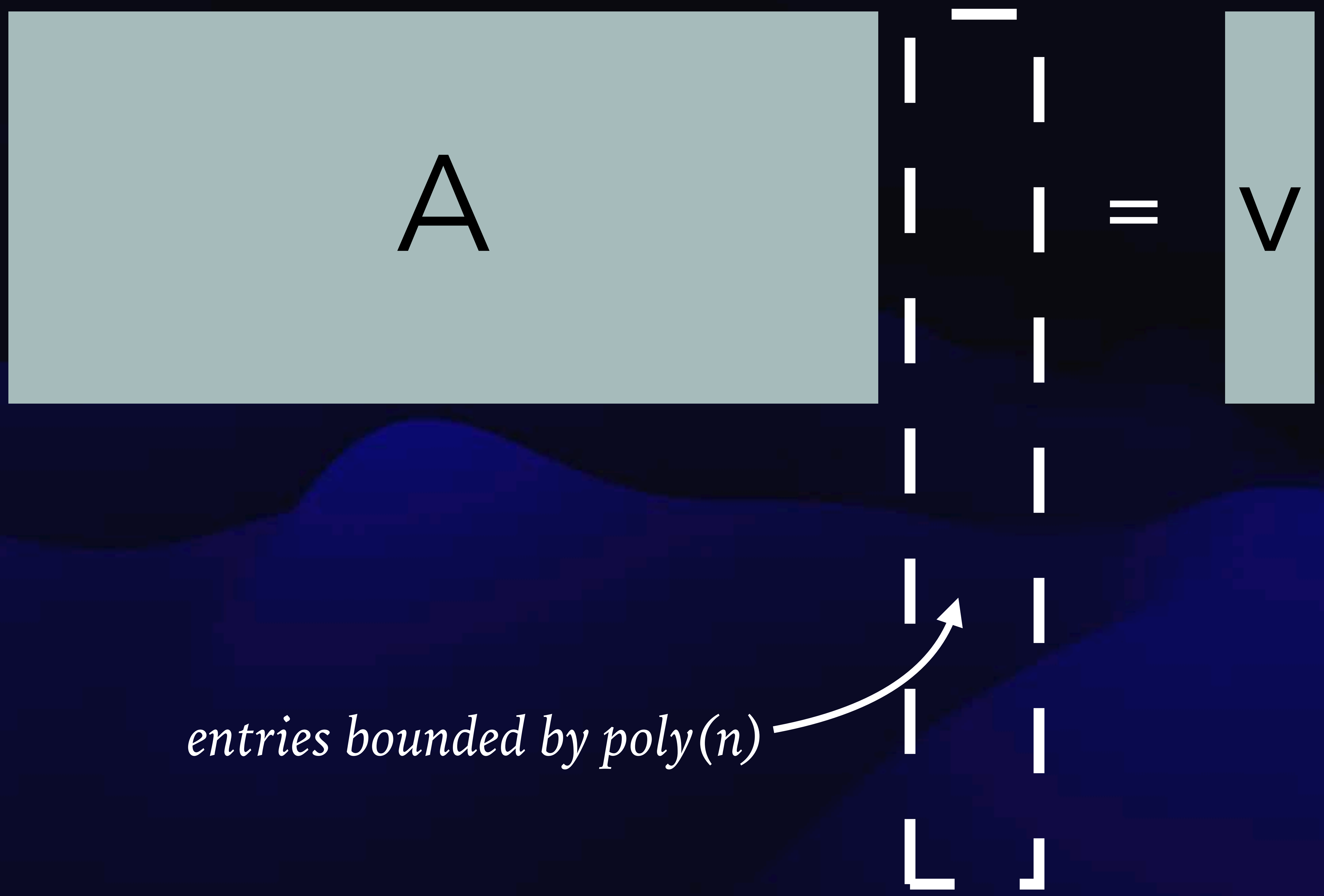
$$\approx_c$$

$$\left\{ \left(\mathbf{A} \quad , \quad \mathbf{u} \right) \ : \ \mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}, \mathbf{u} \leftarrow \mathbb{Z}_q^m \right\}$$

(All computation is mod q, where q is a large modulus)

$$B \approx n, \ \ m \approx n^2, \ \ q \approx 2^{\sqrt{n}}$$

FINDING SHORT PRE-IMAGES



Goal: Given random $\mathbf{A}$, $\mathbf{v}$, find $\mathbf{w}$ with small entries s.t. $\mathbf{A} \cdot \mathbf{w} = \mathbf{v}$.

Qn: Assuming LWE is hard, show that it is hard to find such a $\mathbf{w}$ with small entries for random $\mathbf{A}$.

… but finding short pre-images can be easy if $\mathbf{A}$ is a 'structured' matrix

FINDING SHORT PRE-IMAGES

$$\begin{array}{cccccccccccccccccccc}
1 & 2 & 2^2 & \dots & 2^{\log q} & & & & & & & & & & & & & & & \\
& & & & & 1 & 2 & 2^2 & \dots & 2^{\log q} & & & & & & & & & & \\
& & & & & & & & & & 1 & 2 & 2^2 & \dots & 2^{\log q} & & & & & \\
& & & & & & & & & & & & & & & 1 & 2 & 2^2 & \dots & 2^{\log q}
\end{array}$$

Gadget matrix $\mathbf{G}$

Qn: Given any $\mathbf{v}$, find $\mathbf{w}$ with small entries s.t.

$$\mathbf{G} \cdot \mathbf{w} = \mathbf{v}.$$

FINDING SHORT PRE-IMAGES

$$\left[\begin{array}{c|c} \mathbf{A}' & \mathbf{A}' \cdot \mathbf{R} + \mathbf{G} \end{array}\right]$$

Matrix $\mathbf{A}$

R : square matrix with binary entries

Qn: Given any $\mathbf{v}$, find $\mathbf{w}$ with small entries s.t.

$$\mathbf{A} \cdot \mathbf{w} = \mathbf{v}.$$

FINDING SHORT PRE-IMAGES

Theorem: *It is possible to sample a matrix* $\mathbf{A}$ *with a trapdoor* $T_{\mathbf{A}}$ *s.t.*

- *Using* $T_{\mathbf{A}}$*, we can find pre-image of any* $\mathbf{v}$*.*
- $\mathbf{A}$ *looks like a uniformly random matrix.*
- *If* $\mathbf{v}$ *is uniformly random, then pre-image of* $\mathbf{v}$ *is a random vector with small entries.*

$\mathbf{A}^{-1}(\mathbf{v})$

FINDING SHORT PRE-IMAGES

Extending trapdoor $T_{\mathbf{A}}$ to the right

Qn: Given any $\mathbf{A}$ with trapdoor $T_{\mathbf{A}}$, and $\mathbf{B}, \mathbf{v}$, find $\mathbf{w}$ with small entries s.t.

$$\left[\mathbf{A} \mid \mathbf{B}\right] \cdot \mathbf{w} = \mathbf{v}$$

Extending trapdoor $T_{\mathbf{G}}$ to the left

Qn: Given any $\mathbf{A}$, matrix $\mathbf{R}$ with binary entries, and vector $\mathbf{v}$, find $\mathbf{w}$ with small entries s.t.

$$\left[\mathbf{A} \mid \mathbf{A} \cdot \mathbf{R} + \mathbf{G}\right] \cdot \mathbf{w} = \mathbf{v}$$

Qn (*): Given any $\mathbf{A}, \mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3$ with binary entries, and $\mathbf{v}$, find $\mathbf{w}$ with small entries s.t. $\left[\, \mathbf{A} \mid \mathbf{A} \cdot \mathbf{R}_1 + \mathbf{G} \mid \mathbf{A} \cdot \mathbf{R}_2 + \mathbf{G} \mid \mathbf{A} \cdot \mathbf{R}_3 \,\right] \cdot \mathbf{w} = \mathbf{v}$

SUMMARY OF LATTICE TOOLKIT

$$(\mathbf{A}\ ,\ \mathbf{A}\cdot\mathbf{r}) \approx (\mathbf{A}\ ,\ \mathbf{u})$$

$\mathbf{A}$: *flat uniform matrix* , $\mathbf{r}$: *short entries* , $\mathbf{u}$: *uniform vector*

$$(\mathbf{A}\ ,\ \mathbf{s}\cdot\mathbf{A}+\mathbf{e}) \approx (\mathbf{A}\ ,\ \mathbf{u})$$

$\mathbf{A}$: *flat uniform matrix* , $\mathbf{s}, \mathbf{u}$: *uniform vector* , $\mathbf{e}$: *short entries*

Trapdoor $T_{\mathbf{A}}$ *for matrix* $\mathbf{A}$ *can sample short preimage of any* $\mathbf{v}$

$\mathbf{w}$ *s.t.* $\mathbf{A}\cdot\mathbf{w}=\mathbf{v}$

HOW TO USE LATTICE TOOLKIT FOR CRYPTOGRAPHY

Public Key Encryption

Identity Based Encryption

Attribute Based Encryption

PKE (DUAL-REGEV SCHEME)

$$\text{Setup}() : \mathbf{r} \leftarrow \{0,1\}^m, \quad \mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$$

$$\text{pk} = (\mathbf{A}\ ,\ \mathbf{A}\cdot\mathbf{r}) \quad , \quad \text{sk} = \mathbf{r}$$

$\text{Enc}(\ \text{pk} = (\mathbf{A}, \mathbf{v}), m \in \{0,1\}\ ) :$

Sample $\mathbf{s} \leftarrow \mathbb{Z}_q^n$

$\text{ct}_1 \approx \text{s}^{\text{T}} \cdot \mathbf{A} \qquad \text{ct}_2 \approx \text{s}^{\text{T}} \cdot \mathbf{v} + m \cdot q/2$

Output $(\text{ct}_1, \text{ct}_2)$

Qn: Prove security

$\text{Dec}(\ \text{sk} = \mathbf{r}, (\text{ct}_1, \text{ct}_2)\ ) :$

Compute $z = \text{ct}_2 - \text{ct}_1 \cdot \mathbf{r}$

If z *close to* $q/2$*, output* 1*, else output* 0

HOW TO USE LATTICE TOOLKIT FOR CRYPTOGRAPHY

Public Key Encryption

Identity Based Encryption

Attribute Based Encryption

Solution 1:
Random oracle model

Solution 2:
Standard model

IBE IN THE RANDOM ORACLE MODEL [Gentry-Peikert-Vaikuntanathan 08]

Using mpk *and* ID*, compute a public key* $\mathsf{pk}_{\mathrm{ID}}$ *for ID*

Use Dual-Regev PKE encryption with $\mathsf{pk}_{\mathrm{ID}}$

Using msk *and* ID*, compute secret key* $\mathsf{sk}_{\mathrm{ID}}$ *for ID*

IBE IN THE RANDOM ORACLE MODEL [Gentry-Peikert-Vaikuntanathan 08]

$$H : \mathscr{I}\mathscr{D} \rightarrow \mathbb{Z}_q^n$$

Setup() : mpk $= \mathbf{A}$ msk $= T_{\mathbf{A}}$

Enc(mpk $= \mathbf{A}$, id , $m \in \{0,1\}$) :

$\mathbf{v} = H(\text{id})$ $\text{pk}_{\text{id}} = (\mathbf{A}\ ,\ \mathbf{v})$

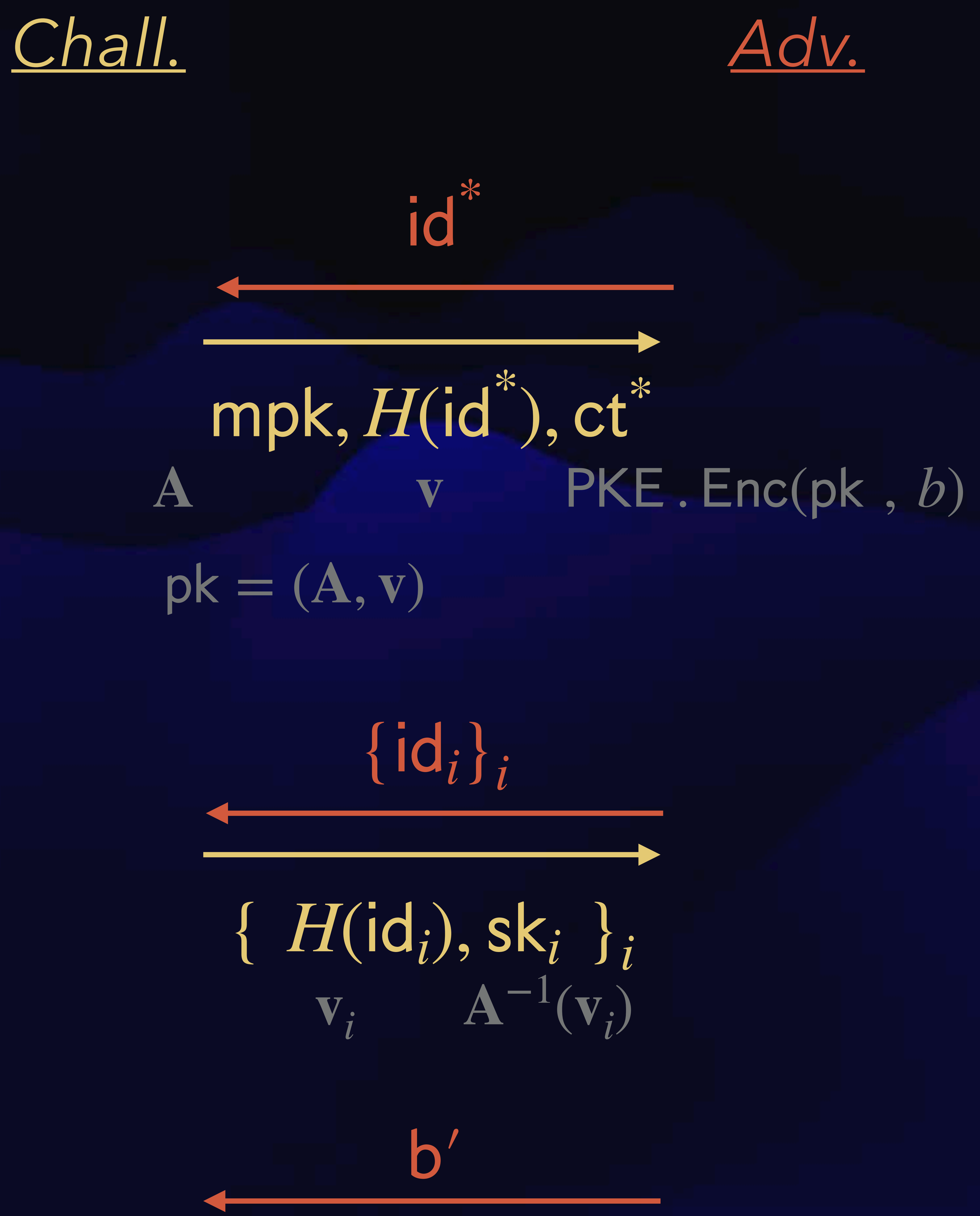
ct $\leftarrow$ PKE . Enc(pk_{id} , m)

KeyGen(msk , id) :

$\mathbf{v} = H(\text{id})$ $\text{sk}_{\text{id}} \leftarrow \mathbf{A}^{-1}(\ \mathbf{v}\ )$

Dec(sk_{id} , ct) : *Output* PKE . Dec(sk_{id} , ct)

SIMPLIFIED SECURITY MODEL AND PROOF



Chall.

Adv.

id^*

$\mathsf{mpk}, H(\mathsf{id}^*), \mathsf{ct}^*$

$\mathbf{A}$ $\mathbf{v}$ $\mathsf{PKE.Enc}(\mathsf{pk}, b)$

$\mathsf{pk} = (\mathbf{A}, \mathbf{v})$

$\{\mathsf{id}_i\}_i$

$\{\ H(\mathsf{id}_i), \mathsf{sk}_i\ \}_i$

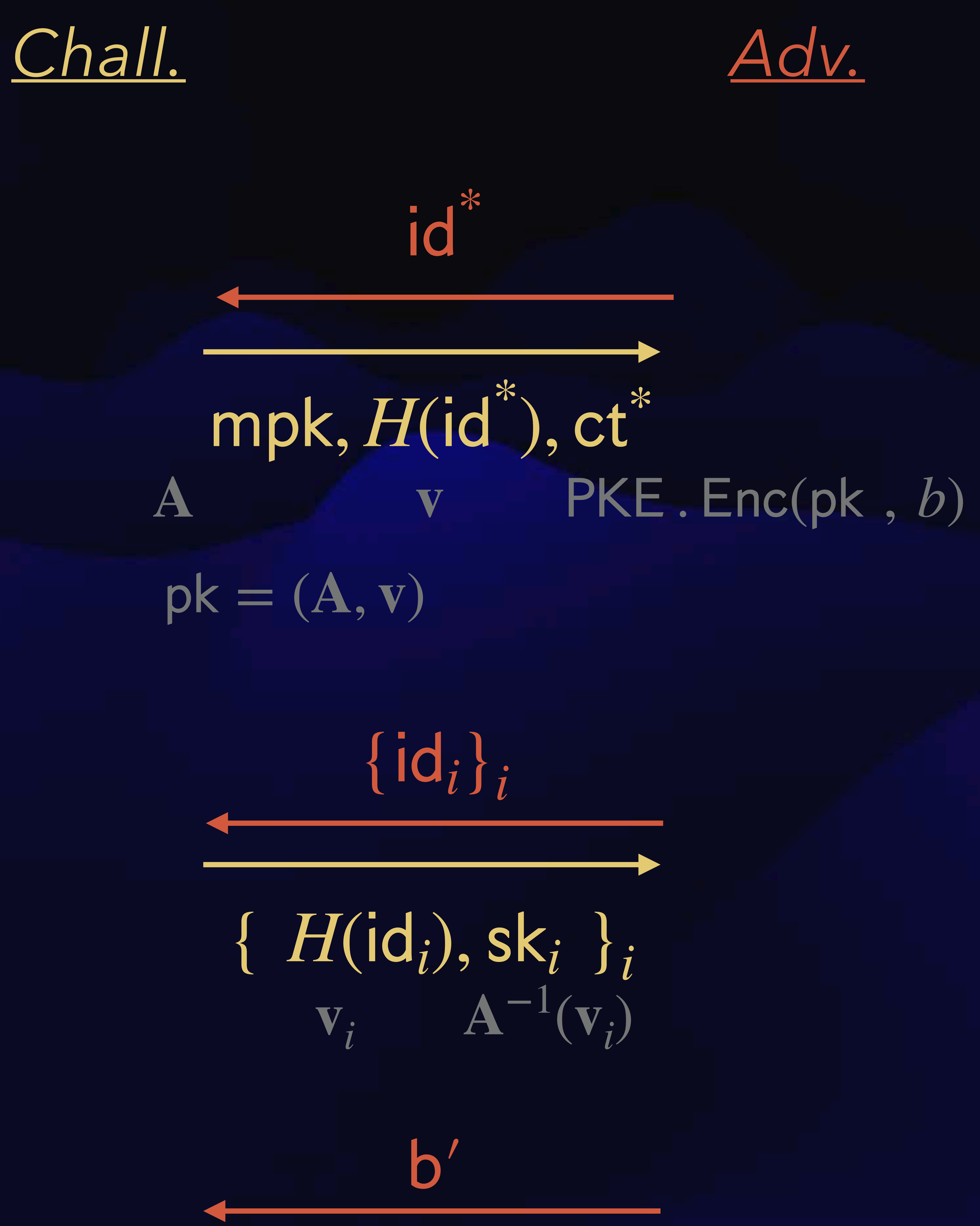
$\mathbf{v}_i$ $\mathbf{A}^{-1}(\mathbf{v}_i)$

b'

- Must use security of PKE scheme
  - Plant the PKE public key and challenge ct′ in the IBE mpk and challenge ciphertext

- Must give out secret keys without knowing $T_{\mathbf{A}}$

SIMPLIFIED SECURITY MODEL AND PROOF



Chall. Adv.

id^*

$\mathsf{mpk}, H(\mathsf{id}^*), \mathsf{ct}^*$

$\mathbf{A}$ $\mathbf{v}$ $\mathsf{PKE.Enc}(\mathsf{pk}, b)$

$\mathsf{pk} = (\mathbf{A}, \mathbf{v})$

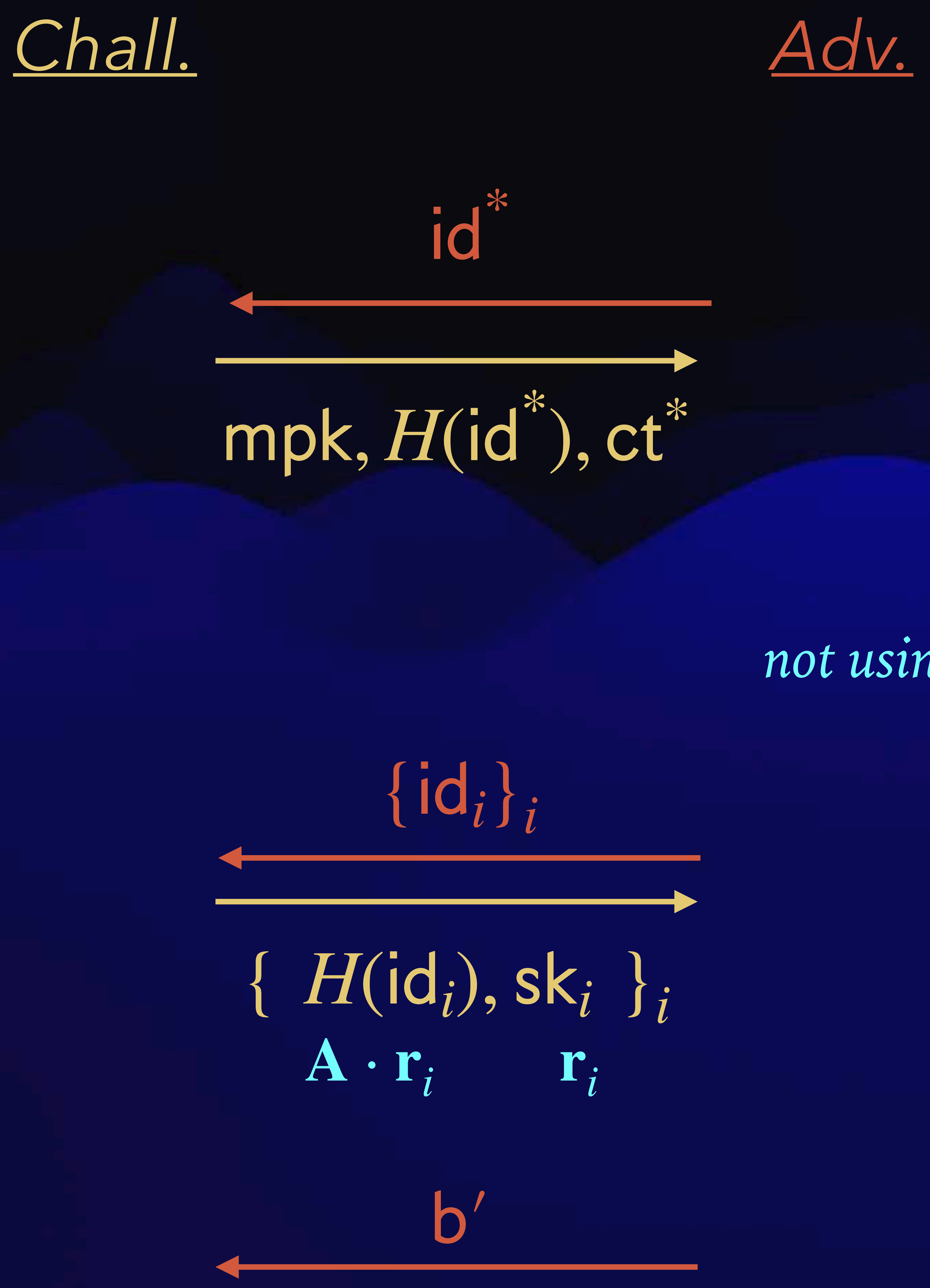
$\{\mathsf{id}_i\}_i$

$\{\ H(\mathsf{id}_i), \mathsf{sk}_i\ \}_i$

$\mathbf{v}_i$ $\mathbf{A}^{-1}(\mathbf{v}_i)$

b'

$$\approx$$



Chall. Adv.

id^*

$\mathsf{mpk}, H(\mathsf{id}^*), \mathsf{ct}^*$

not using $T_{\mathbf{A}}$

$\{\mathsf{id}_i\}_i$

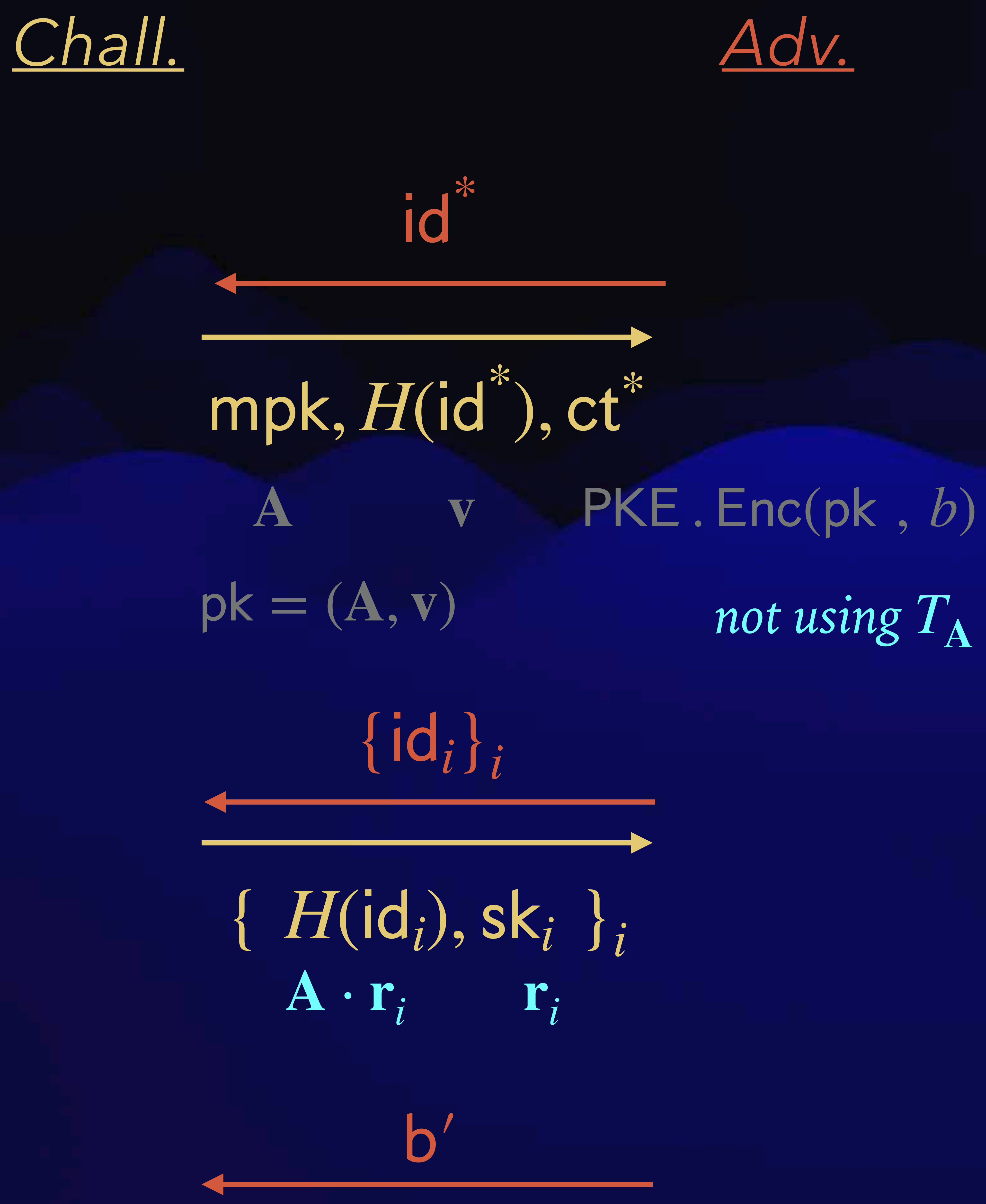
$\{\ H(\mathsf{id}_i), \mathsf{sk}_i\ \}_i$

$\mathbf{A} \cdot \mathbf{r}_i$ $\mathbf{r}_i$

b'

SIMPLIFIED SECURITY MODEL AND PROOF

Can use security of Dual-Regev PKE since only public key used in this experiment



Chall.

Adv.

id^*

$\mathsf{mpk}, H(\mathsf{id}^*), \mathsf{ct}^*$

$\mathbf{A}$ $\mathbf{v}$ $\mathsf{PKE.Enc}(\mathsf{pk}\ ,\ b)$

$\mathsf{pk} = (\mathbf{A}, \mathbf{v})$

not using $T_{\mathbf{A}}$

$\{\mathsf{id}_i\}_i$

$\{\ H(\mathsf{id}_i), \mathsf{sk}_i\ \}_i$

$\mathbf{A} \cdot \mathbf{r}_i$ $\mathbf{r}_i$

b'

IBE IN THE STANDARD MODEL [Cash-Hofheinz-Kiltz-Peikert 10]

Previous construction crucially used the programmability of random oracle.
Construction in the standard model?

Using mpk *and* ID, *compute a public key* $\mathsf{pk}_{\mathrm{ID}}$ *for ID*

Use Dual-Regev PKE encryption with $\mathsf{pk}_{\mathrm{ID}}$

Using msk *and* ID, *compute secret key* $\mathsf{sk}_{\mathrm{ID}}$ *for ID*

IBE IN THE STANDARD MODEL [Cash-Hofheinz-Kiltz-Peikert 10]

$$\mathcal{ID} = \{0,1\}^\ell$$

$$\text{Setup}() : \quad \text{mpk} = \left(\mathbf{A} \ , \ \{\mathbf{A_{i,b}}\}_{i \in [\ell], b \in \{0,1\}} \right) \quad \text{msk} = T_{\mathbf{A}}$$

$$\text{Enc}\left(\ (\ \mathbf{A}, \{\mathbf{A}_{i,b}\} \ ) \ , \text{id} \ , m \in \{0,1\} \ \right) :$$

$$\mathbf{A}_{\text{id}} = \left[\ \mathbf{A} \mid \mathbf{A}_{1,\text{id}_1} \mid \mathbf{A}_{2,\text{id}_2} \mid \ \ldots \ \mid \mathbf{A}_{\ell,\text{id}_\ell} \ \right]$$

$$\text{pk}_{\text{id}} = (\mathbf{A}_{\text{id}} \ , \ \mathbf{v}) \quad \text{ct} \leftarrow \text{PKE.Enc}(\text{pk}_{\text{id}} \ , \ m)$$

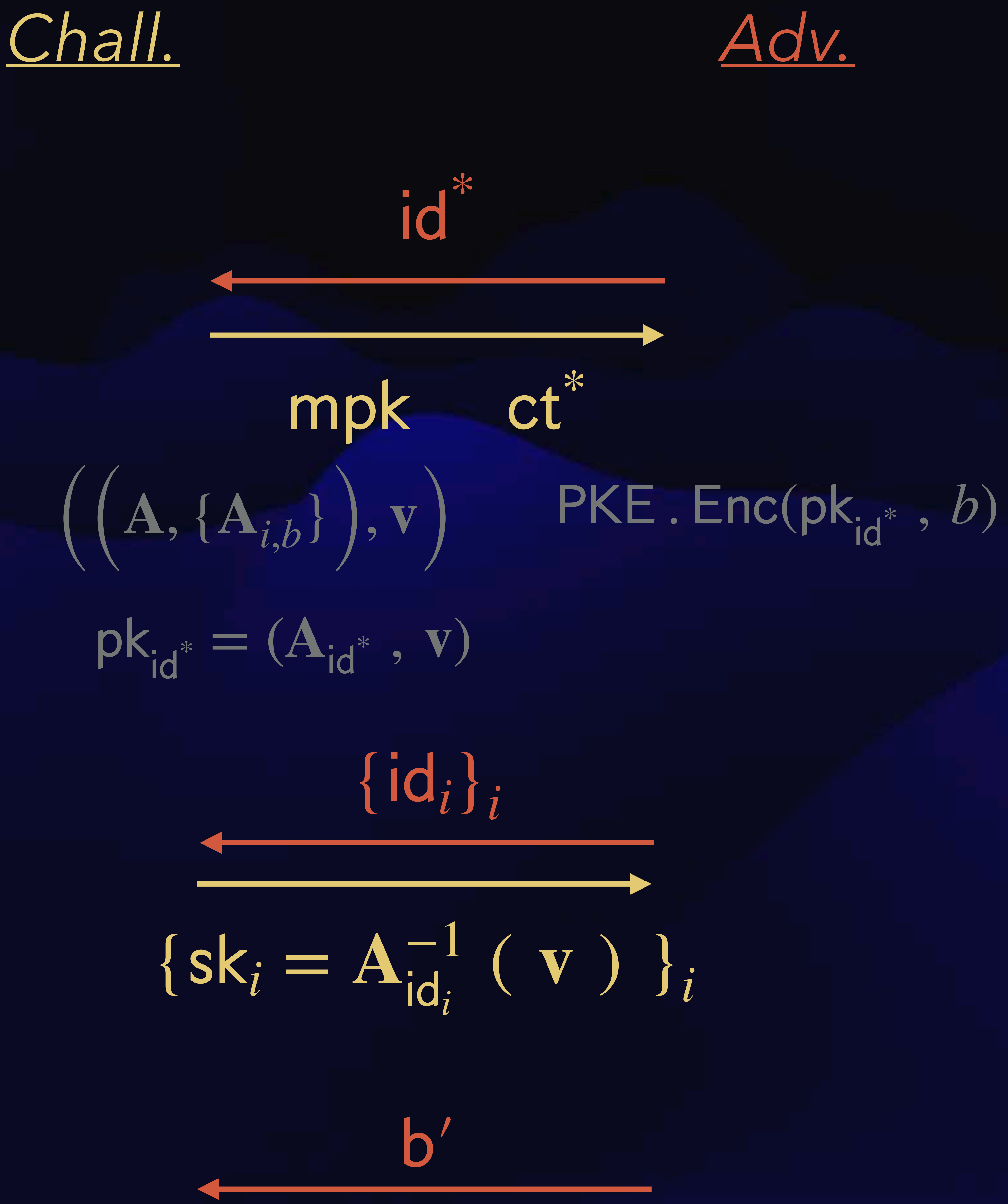
$$\text{KeyGen}(\ T_{\mathbf{A}} \ , \ \text{id} \ ) :$$

Use $T_{\mathbf{A}}$ *to sample* $\mathbf{r} \leftarrow \mathbf{A}_{\text{id}}^{-1}(\mathbf{v})$

$$\text{sk}_{\text{id}} = \mathbf{r}$$

Extending $T_{\mathbf{A}}$
to the right

SIMPLIFIED SECURITY MODEL AND PROOF



Chall. Adv.

id^*

mpk ct*

$\left(\left(\mathbf{A}, \{\mathbf{A}_{i,b}\}\right), \mathbf{v}\right)$ PKE . Enc(pk_{id^*} , b)

$\text{pk}_{\text{id}^*} = (\mathbf{A}_{\text{id}^*} , \mathbf{v})$

$\{\text{id}_i\}_i$

$\{\text{sk}_i = \mathbf{A}_{\text{id}_i}^{-1} (\mathbf{v}) \}_i$

b′

To use security of Dual-Regev PKE,

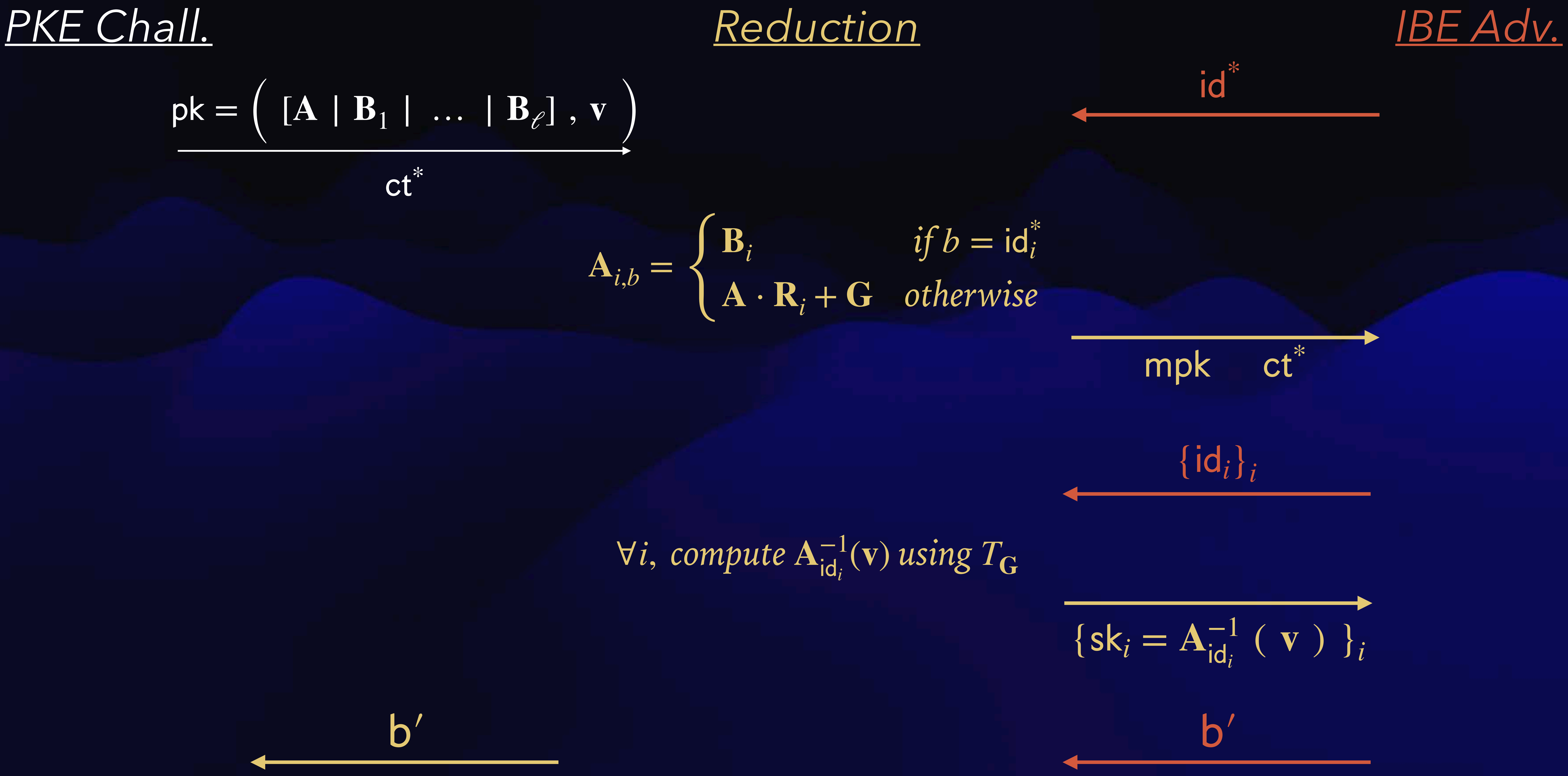
- IBE challenge ct should be the PKE challenge ct
- not use $T_{\mathbf{A}}$ for secret key queries

Idea: set **mpk** s.t. we don't need $T_{\mathbf{A}}$ for sk queries

$$\forall i, b \neq \text{id}_i^*, \ \ \mathbf{A}_{i,b} = \mathbf{A} \cdot \mathbf{R}_i \ + \ \mathbf{G}$$

If id $\neq$ id*, can use $T_{\mathbf{G}}$ to compute sk_{id}

SIMPLIFIED SECURITY MODEL AND PROOF



PKE Chall.

Reduction

IBE Adv.

id^*

$$\xrightarrow[\mathsf{ct}^*]{\mathsf{pk} = \left([\mathbf{A} \mid \mathbf{B}_1 \mid \ \ldots \ \mid \mathbf{B}_\ell] \,, \mathbf{v} \right)}$$

$$\mathbf{A}_{i,b} = \begin{cases} \mathbf{B}_i & \textit{if } b = \mathsf{id}_i^* \\ \mathbf{A} \cdot \mathbf{R}_i + \mathbf{G} & \textit{otherwise} \end{cases}$$

$\mathsf{mpk} \quad \mathsf{ct}^*$

$\{\mathsf{id}_i\}_i$

$\forall i,$ *compute* $\mathbf{A}_{\mathsf{id}_i}^{-1}(\mathbf{v})$ *using* $T_{\mathbf{G}}$

$\{\mathsf{sk}_i = \mathbf{A}_{\mathsf{id}_i}^{-1} (\ \mathbf{v} \ ) \ \}_i$

b'

b'

HOW TO USE LATTICE TOOLKIT FOR CRYPTOGRAPHY

Public Key Encryption

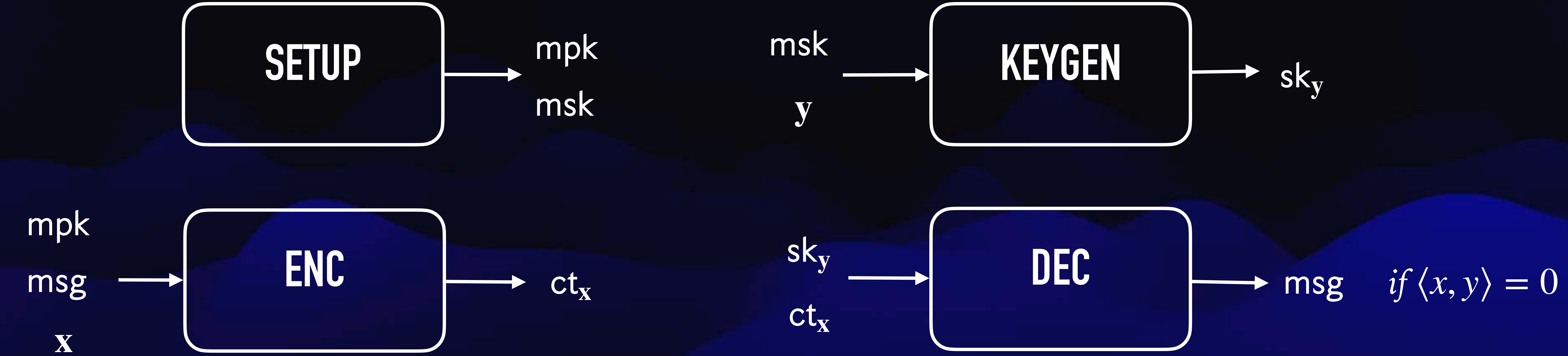
Identity Based Encryption

Attribute Based Encryption

Solution for Inner product policy

ABE FOR INNER-PRODUCTS [Agrawal-Freeman-Vaikuntanathan 11]

Attribute space = Policy space = $[-T, T]^{\ell}$ for some constant T

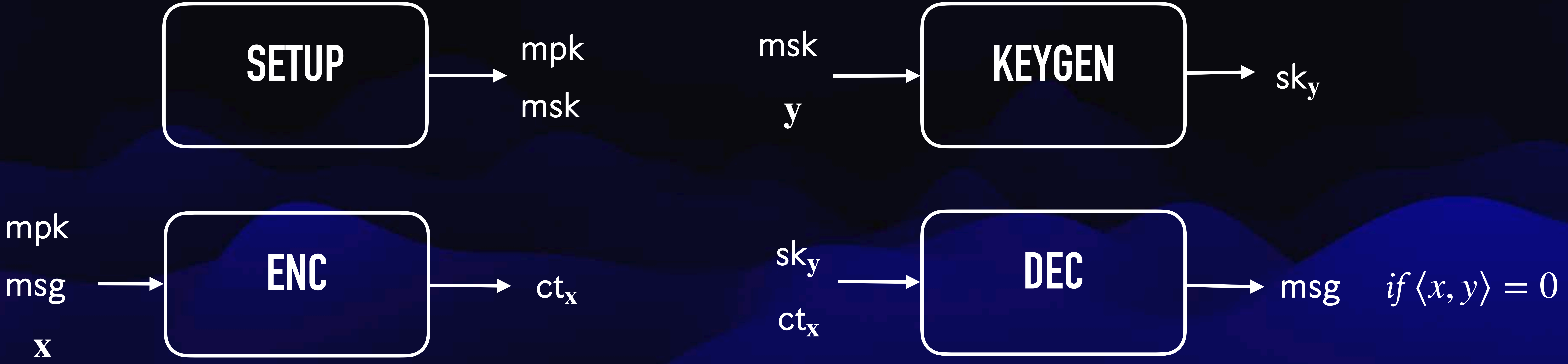

SETUP
mpk
msk
msk
$\mathbf{y}$
KEYGEN
$\mathsf{sk}_{\mathbf{y}}$
mpk
msg
$\mathbf{x}$
ENC
$\mathsf{ct}_{\mathbf{x}}$
$\mathsf{sk}_{\mathbf{y}}$
$\mathsf{ct}_{\mathbf{x}}$
DEC
msg
if $\langle x, y \rangle = 0$

Why inner products?

Inner products capture expressive policies such as polynomial eval, CNFs, DNFs, etc.

ABE FOR INNER-PRODUCTS [Agrawal-Freeman-Vaikuntanathan 11]

Attribute space = Policy space = $[-T, T]^\ell$ for some constant T


SETUP
mpk
msk
msk
$\mathbf{y}$
KEYGEN
$\mathsf{sk}_{\mathbf{y}}$
mpk
msg
$\mathbf{x}$
ENC
$\mathsf{ct}_{\mathbf{x}}$
$\mathsf{sk}_{\mathbf{y}}$
$\mathsf{ct}_{\mathbf{x}}$
DEC
msg
if $\langle x, y \rangle = 0$

Why inner products?

Inner products capture expressive policies such as polynomial eval, CNFs, DNFs, etc.

Previous idea of directly 'plugging in' Dual-Regev PKE does not work here :(

ABE FOR INNER-PRODUCTS [Agrawal-Freeman-Vaikuntanathan 11]

$$\text{Setup}(): \quad \text{mpk} = \left(\mathbf{A}\ ,\ \{\mathbf{A_i}\}_{i\in[\ell]}\right) \quad \text{msk} = T_{\mathbf{A}}$$

$$\text{Enc}\Big(\ (\ \mathbf{A}, \{\mathbf{A}_i\}\ )\ ,\ \mathbf{x}\ ,\ m \in \{0,1\}\ \Big):$$

$$\mathbf{A_x} = \Big[\ \mathbf{A} \mid \mathbf{A}_1 + \mathbf{x}_1 \cdot \mathbf{G} \mid\ \dots\ \mid \mathbf{A}_\ell + \mathbf{x}_\ell \cdot \mathbf{G}\ \Big]$$

$$\text{pk}_{\mathbf{x}} = (\mathbf{A_x}\ ,\ \mathbf{v}) \quad \text{ct} \leftarrow \text{PKE}.\text{Enc}(\text{pk}_{\mathbf{x}}\ ,\ m)$$

Qn: How to decrypt?

Ans: $\mathbf{A_x} \rightarrow \left[\mathbf{A} \mid \Sigma\ y_i\mathbf{A}_i\right] = \mathbf{B_y}$, ct = $(\text{ct}_{\mathbf{x}}, \text{ct}_{\mathbf{v}})$

$\text{ct}_{\mathbf{x}} \rightarrow \text{PKE}.\text{Enc}(\ \mathbf{B_y}, m\ ) = \text{ct}_{\mathbf{y}}$

Compute $\text{ct}_{\mathbf{v}}\ -\ \text{ct}_{\mathbf{y}} \cdot \text{sk}$

$$\text{KeyGen}(\ T_{\mathbf{A}}\ ,\ \mathbf{y}\ ): \mathbf{B_y} = \Big[\ \mathbf{A} \mid \Sigma_i\ \mathbf{y}_i \cdot \mathbf{A}_i\ \Big]$$

Use $T_{\mathbf{A}}$ to sample $\mathbf{r} \leftarrow \mathbf{B_y}^{-1}(\mathbf{v})$

$$\text{sk}_{\text{id}} = \mathbf{r}$$

ABE FOR GENERAL CIRCUITS [Boneh-Gentry-Gorbunov-Halevi-Nikolaenko-Segev-Vaikuntanathan-Vinayagamurthy 14]

$$\text{Setup}() : \quad \text{mpk} = \left(\mathbf{A}\ ,\ \{\mathbf{A_i}\}_{i \in [\ell]}\right) \quad \text{msk} = T_{\mathbf{A}}$$

Structure very similar to inner-products construction

$$\text{Enc}\Big(\ (\ \mathbf{A}, \{\mathbf{A}_i\}\ )\ ,\ \mathbf{x}\ ,\ m \in \{0,1\}\ \Big) :$$

$$\mathbf{A_x} = \big[\ \mathbf{A} \mid \mathbf{A}_1 + \mathbf{x}_1 \cdot \mathbf{G} \mid \ \ldots \ \mid \mathbf{A}_\ell + \mathbf{x}_\ell \cdot \mathbf{G}\ \big]$$

$$\text{pk}_{\mathbf{x}} = (\mathbf{A_x}\ ,\ \mathbf{v}) \quad \text{ct} \leftarrow \text{PKE}.\text{Enc}(\text{pk}_{\mathbf{x}}\ ,\ m)$$

$$\text{ct} = (\text{ct}_{\mathbf{x}}, \text{ct}_{\mathbf{v}})$$

$$\text{ct}_{\mathbf{x}} \rightarrow \mathbf{s}^{\mathrm{T}} \cdot \mathbf{B}_f + f(x)\mathbf{G} + \ \textit{noise}$$

$$\text{KeyGen}(\ T_{\mathbf{A}}\ , f\ ) : \mathbf{B}_f = \big[\ \mathbf{A} \mid \mathbf{A}_f\ \big]$$

Use $T_{\mathbf{A}}$ *to sample* $\mathbf{r} \leftarrow \mathbf{B}_f^{-1}(\mathbf{v})$

$$\text{sk}_{\text{id}} = \mathbf{r}$$

CONCLUSIONS

- [Gentry-Peikert-Vaikuntanathan 08] : First lattice-based IBE scheme in the random oracle model.

- [Cash-Hofheinz-Kiltz-Peikert 10]: Lattice based IBE in the standard model. Later, a more efficient lattice-based construction was given by [Agrawal-Boneh-Boyen 11]

- [Agrawal-Freeman-Vaikuntanathan 11]: Lattice based ABE for inner-products

- [Gorbunov-Vaikuntanathan-Wee 13]: First lattice based construction for all circuits. An improved construction was given by [Boneh-Gentry-Gorbunov-Halevi-Nikolaenko-Segev-Vaikuntanathan-Vinayagamurthy 14].

- Several improvements over the last few years. ABE where policies can be described using finite automata, Turing machines, etc.

THANK YOU !