FAME: Fast Attribute-based Message Encryption

Shashank Agrawal

VISA Research

Melissa Chase

Microsoft Research

Attribute-based Encryption

- Applications in a variety of settings:
  - network privacy [BBSBS09], pay-per-view broadcasting [TBEM08], health-record access control [APGLPR11, CDEN12], cloud security [SRGS12, verifiable computation [PRV12], forward-secure messaging [GM15], easy-to-use secure email [RAHZS16], ...

Attribute-based Encryption

- Applications in a variety of settings:
  - network privacy [BBSBS09], pay-per-view broadcasting [TBEM08], health-record access control [APGLPR11, CDEN12], cloud security [SRGS12, verifiable computation [PRV12], forward-secure messaging [GM15], easy-to-use secure email [RAHZS16], ...
- Not a surprise: Fine-grained control

Impact

- Limited impact on the real-world

Impact

- Limited impact on the real-world
- Central issues:
  - Strong security guarantee
  - Fast operations
  - Desirable features

New Schemes!

New Schemes!

- Simultaneously:

New Schemes!

- Simultaneously:
  - No restriction on size of policies or attribute sets

New Schemes!

- Simultaneously:
  - No restriction on size of policies or attribute sets
  - Any arbitrary string can be used as an attribute

New Schemes!

- Simultaneously:
  - No restriction on size of policies or attribute sets
  - Any arbitrary string can be used as an attribute
  - Based on Type-III pairing groups

New Schemes!

- Simultaneously:
  - No restriction on size of policies or attribute sets
  - Any arbitrary string can be used as an attribute
  - Based on Type-III pairing groups
  - Small number of pairings for decryption

New Schemes!

- Simultaneously:
  - No restriction on size of policies or attribute sets
  - Any arbitrary string can be used as an attribute
  - Based on Type-III pairing groups
  - Small number of pairings for decryption
  - Satisfy the natural security requirement

New Schemes!

- Simultaneously:
  - No restriction on size of policies or attribute sets
  - Any arbitrary string can be used as an attribute
  - Based on Type-III pairing groups
  - Small number of pairings for decryption
  - Satisfy the natural security requirement
    - under a standard hardness assumption + random oracle

New Schemes!

- Simultaneously:
  - No restriction on size of policies or attribute sets
  - Any arbitrary string can be used as an attribute
  - Based on Type-III pairing groups
  - Small number of pairings for decryption
  - Satisfy the natural security requirement
    - under a standard hardness assumption + random oracle
- **Improve upon popular and state-of-the-art schemes in several ways**

Time to Upgrade!

- Ciphertext-policy ABE
  - Bethencourt, Sahai, and Waters, IEEE S&P, 2007
  - Our scheme: more secure, faster, lighter

ABE, *formally*

Attributes & Policies

Attributes & Policies

Attribute: property

Attributes & Policies

Attribute: property

Policy: Boolean expression on attributes

Attributes & Policies

Attribute: property

Policy: Boolean expression on attributes

Zipcode:90240

AgeGroup:Over65

City:MountainView

Attributes & Policies

<u>Attribute: property</u>

<u>Policy: Boolean expression on attributes</u>

Zipcode:90240

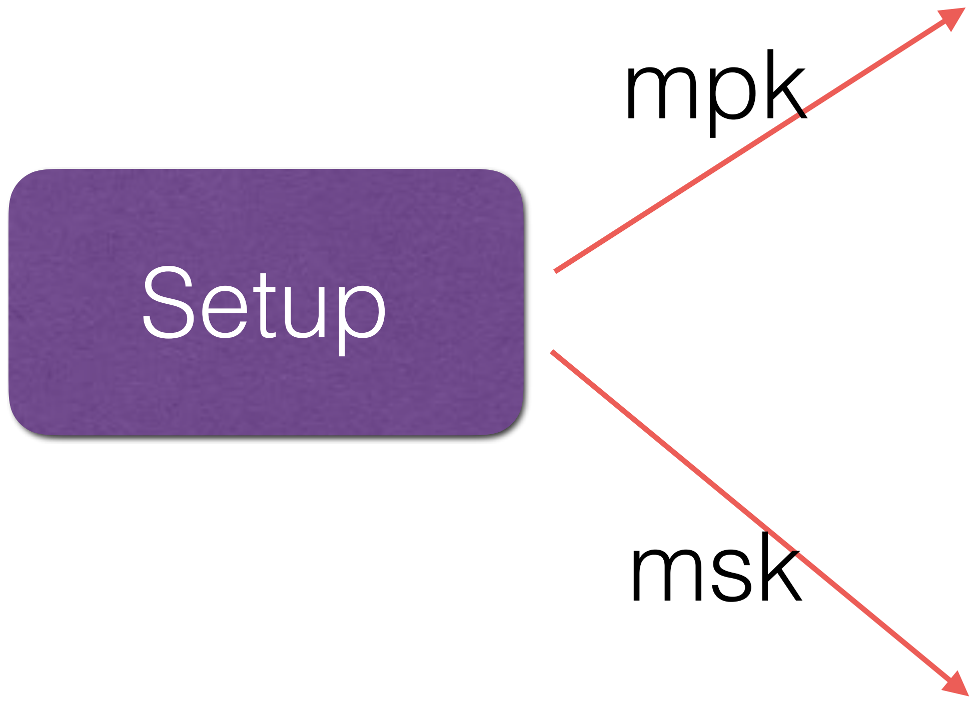
AgeGroup:Over65

City:MountainView

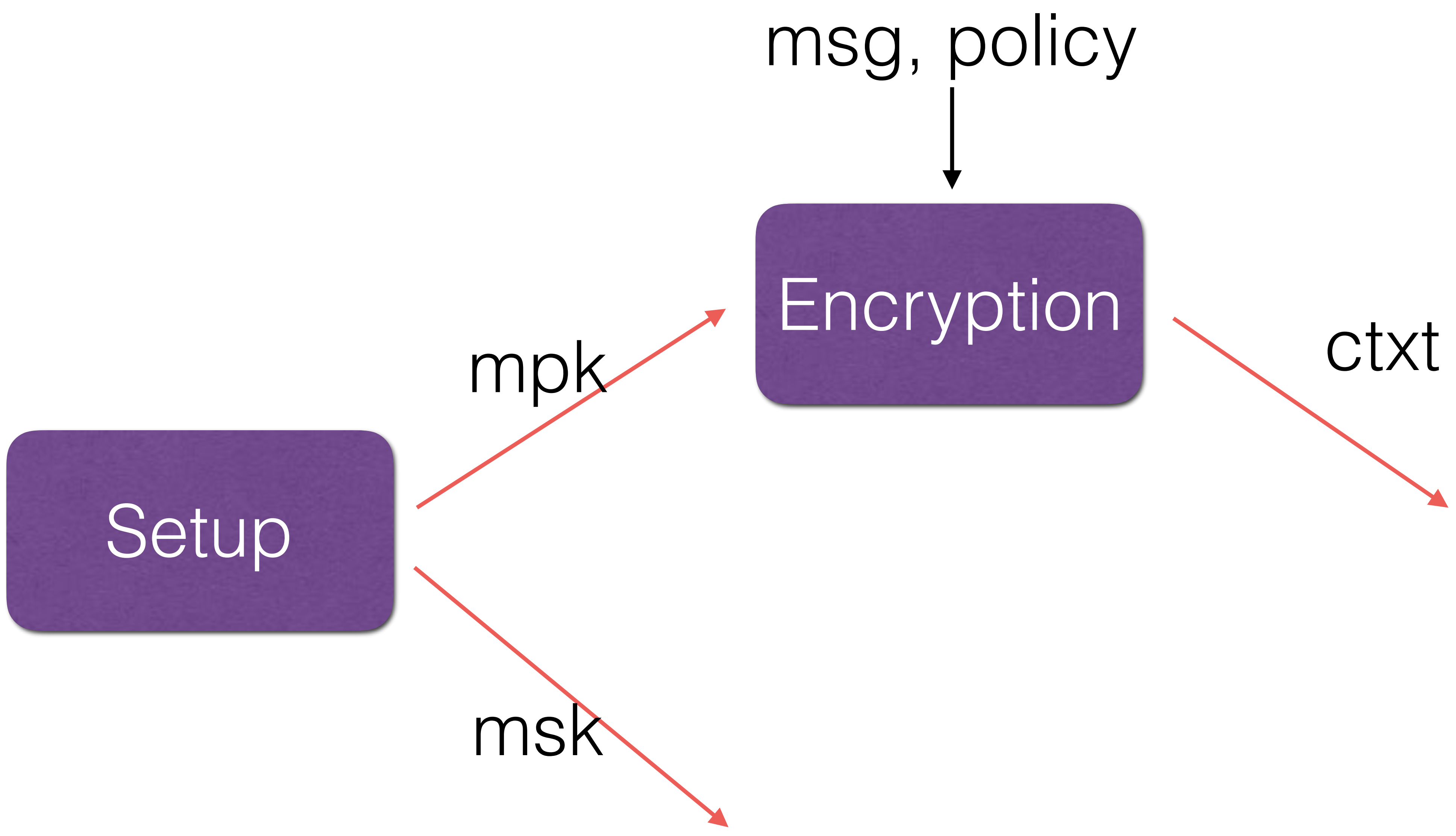
(Zipcode:90240 OR City:BeverlyHills)
AND (AgeGroup:18-25)

The constituents

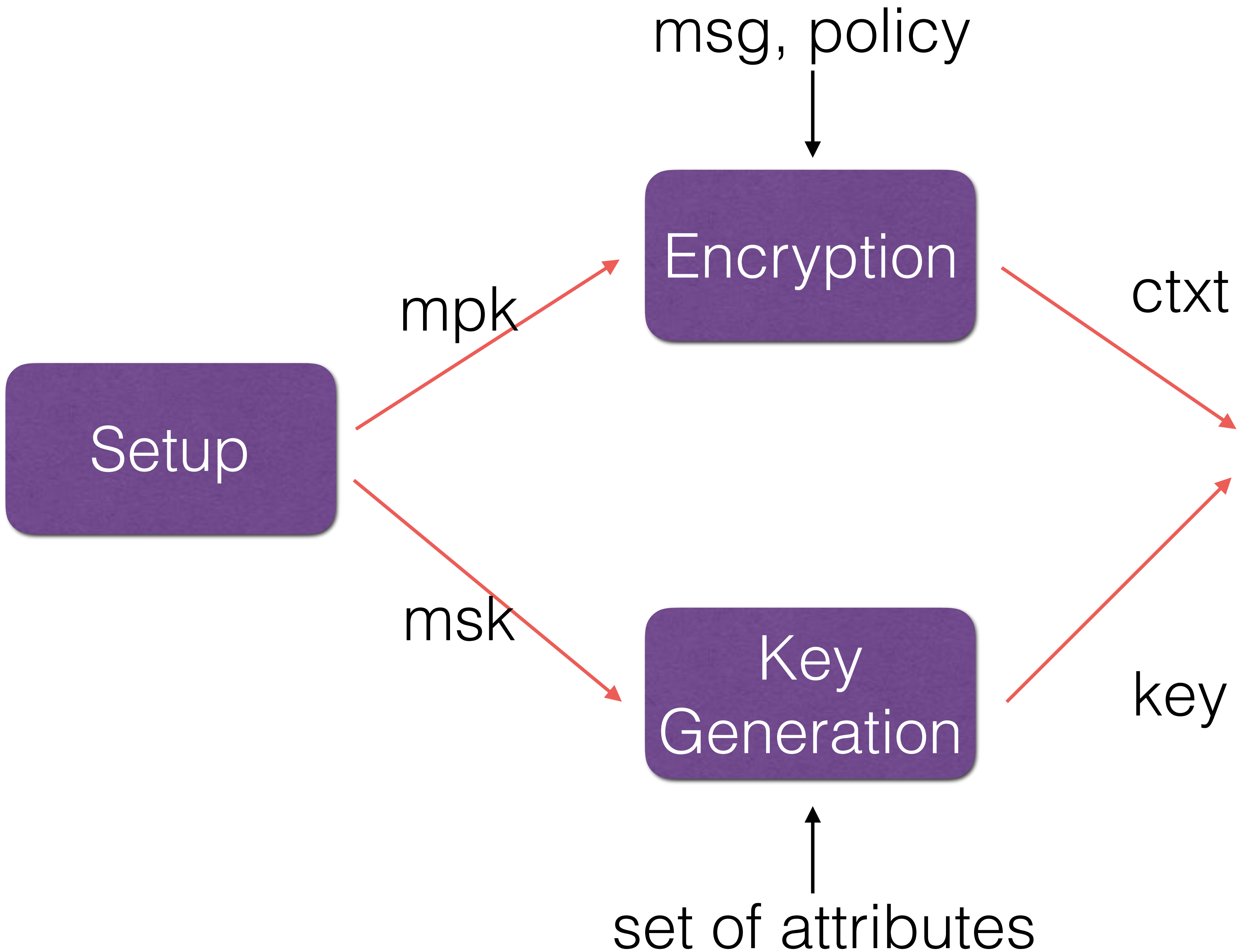
The constituents



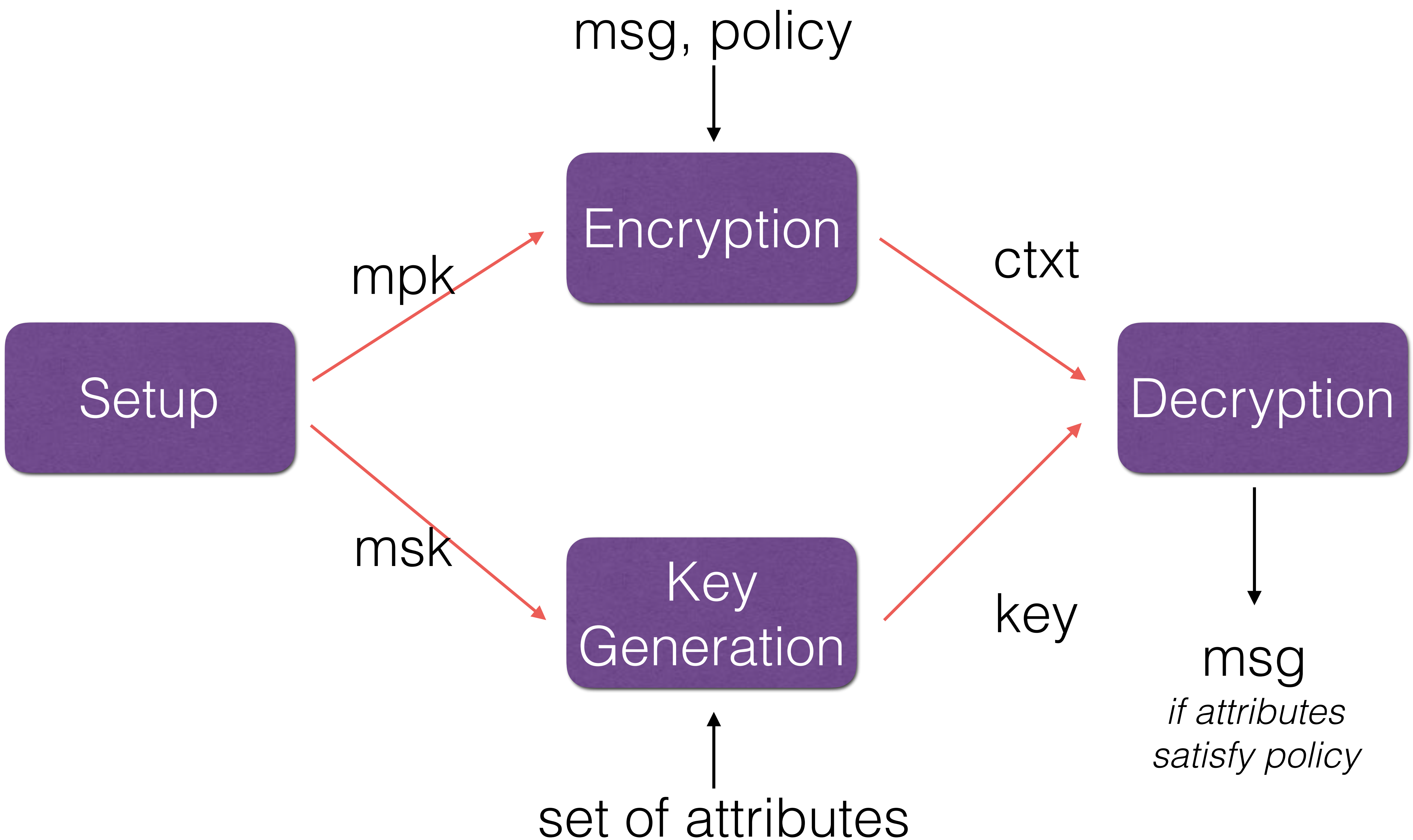
The constituents



The constituents



The constituents



Properties, *we desire*

Unrestricted Policies & Attribute Sets

- As institutions grow, more and more complex roles, entities, policies, procedures, etc.

Unrestricted Policies & Attribute Sets

- As institutions grow, more and more complex roles, entities, policies, procedures, etc.
- ABE schemes restrict policies, attributes

Unrestricted Policies & Attribute Sets

- As institutions grow, more and more complex roles, entities, policies, procedures, etc.
- ABE schemes restrict policies, attributes
- Bounds: number of attributes in a key, size of access policies in ciphertexts

Unrestricted Policies & Attribute Sets

- As institutions grow, more and more complex roles, entities, policies, procedures, etc.
- ABE schemes restrict policies, attributes
- Bounds: number of attributes in a key, size of access policies in ciphertexts
- Problems:
  - Limit expressiveness

Unrestricted Policies & Attribute Sets

- As institutions grow, more and more complex roles, entities, policies, procedures, etc.
- ABE schemes restrict policies, attributes
- Bounds: number of attributes in a key, size of access policies in ciphertexts
- Problems:
  - Limit expressiveness
  - Adversely effect the behavior: generous vs tight

Unrestricted Policies & Attribute Sets

- As institutions grow, more and more complex roles, entities, policies, procedures, etc.
- ABE schemes restrict policies, attributes
- Bounds: number of attributes in a key, size of access policies in ciphertexts
- Problems:
  - Limit expressiveness
  - Adversely effect the behavior: generous vs tight
- Our schemes: No restriction on size of attributes sets & policies

Attribute Usage

- Kind and number of attributes

Attribute Usage

- Kind and number of attributes
- Policy: (Zipcode:90240 OR City:BeverlyHills) AND (AgeGroup:18-25)

Attribute Usage

- Kind and number of attributes
- Policy: (Zipcode:90240 OR City:BeverlyHills) AND (AgeGroup:18-25)
- Issue keys for **every** zip-code and city

Attribute Usage

- Kind and number of attributes
- Policy: (Zipcode:90240 OR City:BeverlyHills) AND (AgeGroup:18-25)
- Issue keys for **every** zip-code and city
- ABE schemes are small universe

Attribute Usage

- Kind and number of attributes
- Policy: (Zipcode:90240 OR City:BeverlyHills) AND (AgeGroup:18-25)
- Issue keys for **every** zip-code and city
- ABE schemes are small universe
    - A-priori bound on number of different attributes

Attribute Usage

- Kind and number of attributes
- Policy: (Zipcode:90240 OR City:BeverlyHills) AND (AgeGroup:18-25)
- Issue keys for **every** zip-code and city
- ABE schemes are small universe
    - A-priori bound on number of different attributes
    - Size of public-key scales linearly

Attribute Usage

- Kind and number of attributes
- Policy: (Zipcode:90240 OR City:BeverlyHills) AND (AgeGroup:18-25)
- Issue keys for **every** zip-code and city
- ABE schemes are small universe
  - A-priori bound on number of different attributes
  - Size of public-key scales linearly
- 43000 zip-codes, 20000 cities

Attribute Usage

- Kind and number of attributes
- Policy: (Zipcode:90240 OR City:BeverlyHills) AND (AgeGroup:18-25)
- Issue keys for **every** zip-code and city
- ABE schemes are small universe
  - A-priori bound on number of different attributes
  - Size of public-key scales linearly
- 43000 zip-codes, 20000 cities
- Names, addresses? Hundreds of millions, grow rapidly

Attribute Usage

- Kind and number of attributes
- Policy: (Zipcode:90240 OR City:BeverlyHills) AND (AgeGroup:18-25)
- Issue keys for **every** zip-code and city
- ABE schemes are small universe
    - A-priori bound on number of different attributes
    - Size of public-key scales linearly
- 43000 zip-codes, 20000 cities
- Names, addresses? Hundreds of millions, grow rapidly
- Our schemes: **arbitrary string** can be an attribute

Pairings

Pairings

- Triple of groups $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T)$

Pairings

- Triple of groups $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T)$
- Map $e : \mathbb{G} \times \mathbb{H} \rightarrow \mathbb{G}_T$

Pairings

- Triple of groups $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T)$
- Map $e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$
- Options:

Pairings

- Triple of groups $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T)$
- Map $e : \mathbb{G} \times \mathbb{H} \rightarrow \mathbb{G}_T$
- Options:
  - Composite-order: large representation, slow pairings

Pairings

- Triple of groups $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T)$
- Map $e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$
- Options:
  - Composite-order: large representation, slow pairings
  - Prime-order symmetric: security issues

Pairings

- Triple of groups $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T)$
- Map $e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$
- Options:
  - Composite-order: large representation, slow pairings
  - Prime-order symmetric: security issues
  - Prime-order asymmetric (Type-III)

Decryption

Decryption

- Most important procedure
  - Computationally weak devices

Decryption

- Most important procedure
  - Computationally weak devices
- Initial work: linear in the number of attributes

Decryption

- Most important procedure
  - Computationally weak devices
- Initial work: linear in the number of attributes
- Our work: 6 pairing operations

Security

Security

- Natural requirement: Full/Adaptive security
  - Attack policy chosen adaptively

Security

- Natural requirement: Full/Adaptive security
  - Attack policy chosen adaptively
- Unrealistic: Selective security
  - Policy declared upfront

Security

- Natural requirement: Full/Adaptive security
  - Attack policy chosen adaptively
- Unrealistic: Selective security
  - Policy declared upfront
- Hardness assumption
  - Decisional linear (DLIN) vs q-type

Designing, *our schemes*

Represent Policies

Represent Policies

- Boolean formulae: ANDs, ORs

Represent Policies

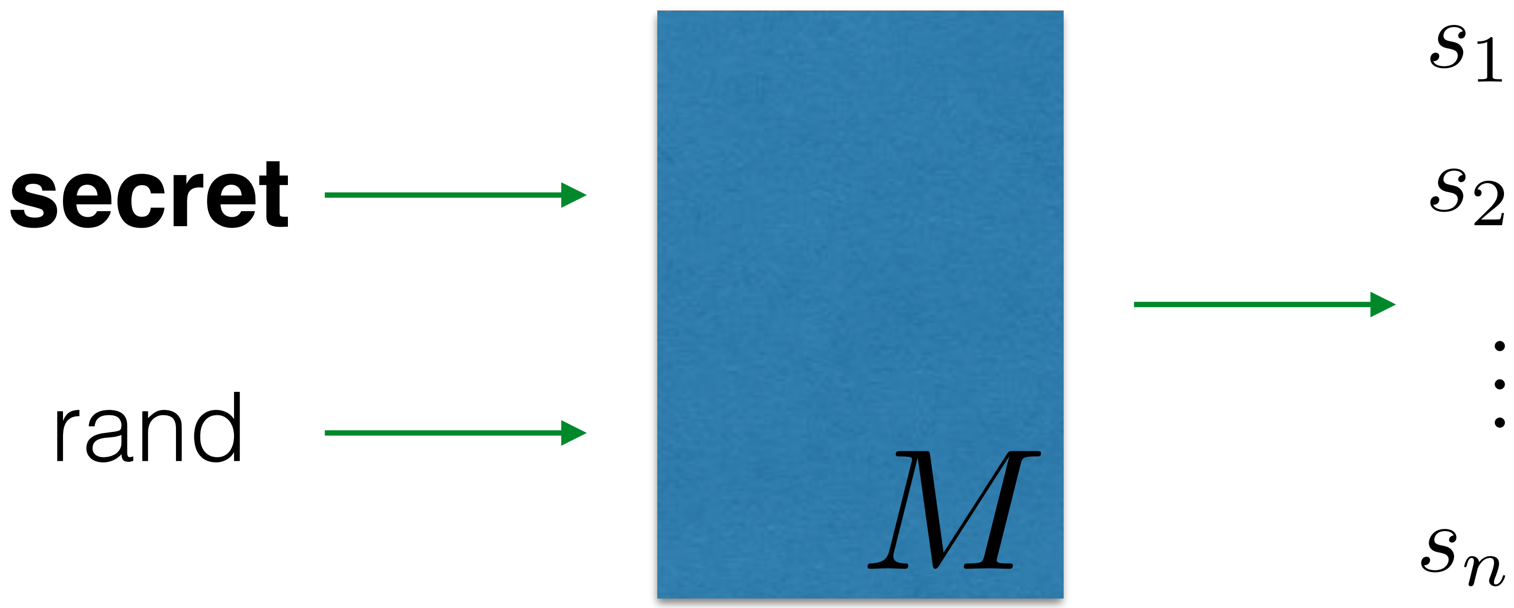
- Boolean formulae: ANDs, ORs
- Monotone span programs (M, π)

Represent Policies

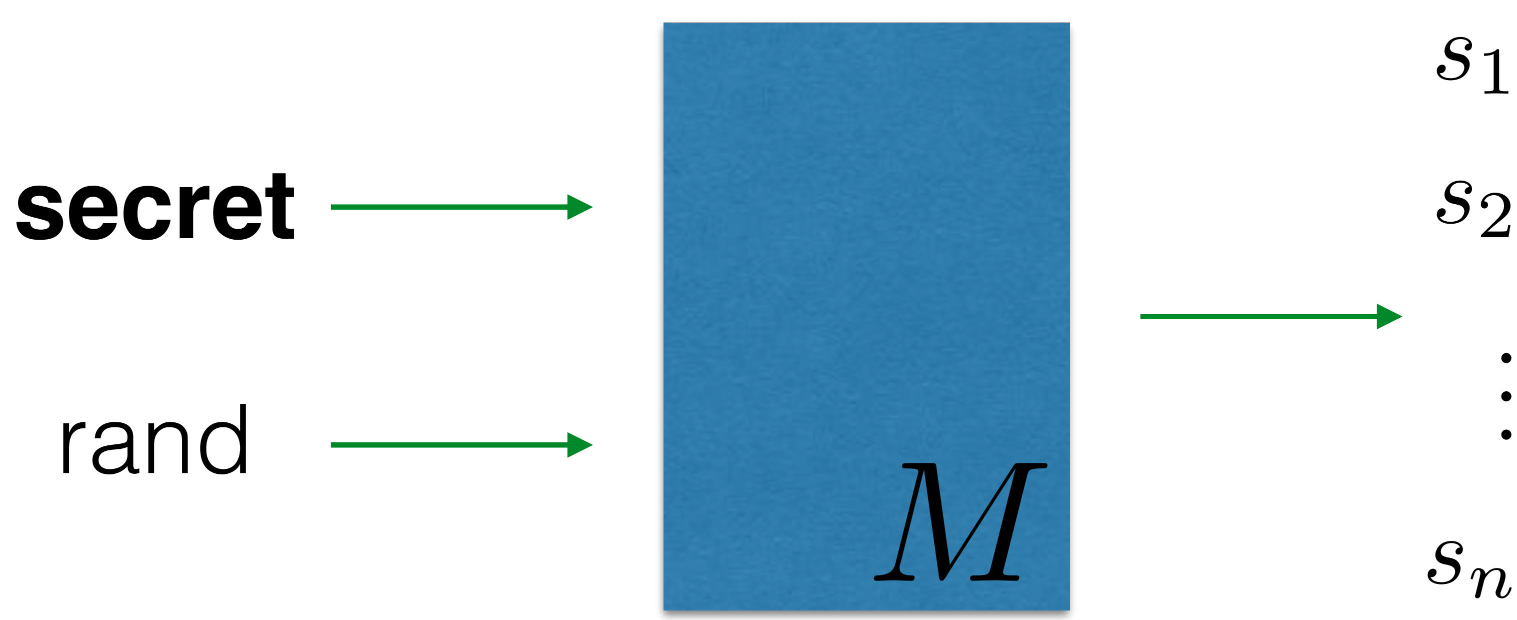
- Boolean formulae: ANDs, ORs
- Monotone span programs (M, π)

$$\left[\begin{array}{c} \\ \\ \\ \\ M \end{array}\right] \begin{array}{l} \xrightarrow{\pi} \mathsf{attr}_1 \\ \longrightarrow \mathsf{attr}_2 \\ \\ \\ \longrightarrow \mathsf{attr}_n \end{array}$$

Monotone Span Programs



Monotone Span Programs



set of attributes

$S = \{\mathsf{attr}_1, \mathsf{attr}_5\}$ satisfies (M, π) $\xrightarrow[s_1,\, s_5]{\text{linearly combine}}$ **secret**

CP-ABE: High-level Design

CP-ABE: High-level Design

set of attributes

$S = \{\mathsf{attr}_1, \mathsf{attr}_5\}$

CP-ABE: High-level Design

set of attributes

$$S = \{\mathsf{attr}_1, \mathsf{attr}_5\}$$

$$\mathsf{key} = (\mathsf{sk}_1, \mathsf{sk}_5)$$

CP-ABE: High-level Design

set of attributes

$S = \{\mathsf{attr}_1, \mathsf{attr}_5\}$

msg

$\mathsf{key} = (\mathsf{sk}_1, \mathsf{sk}_5)$

CP-ABE: High-level Design

set of attributes

$S = \{\mathsf{attr}_1, \mathsf{attr}_5\}$

$\mathsf{key} = (\mathsf{sk}_1, \mathsf{sk}_5)$

msg **secret**

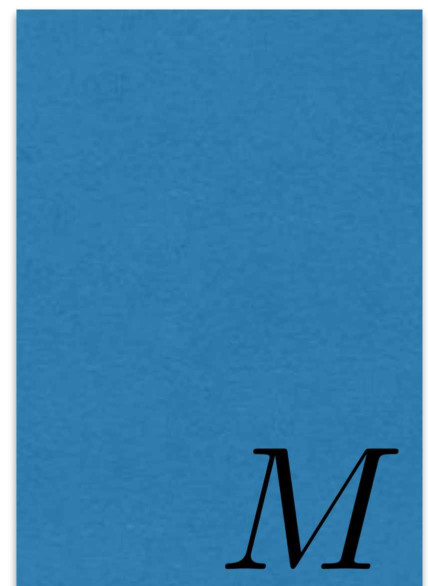
CP-ABE: High-level Design

set of attributes

$S = \{\mathsf{attr}_1, \mathsf{attr}_5\}$

$\mathsf{key} = (\mathsf{sk}_1, \mathsf{sk}_5)$

msg **secret**



s_1

s_2

$\vdots$

s_n

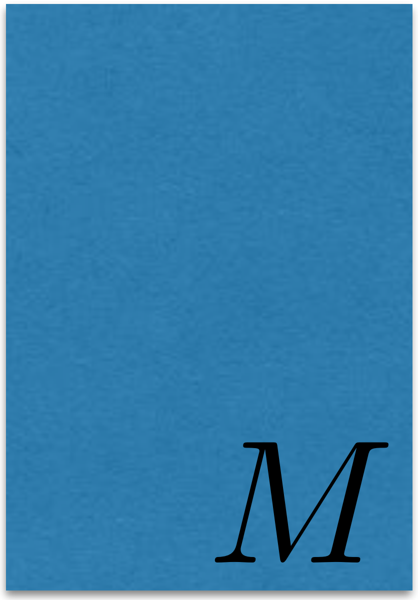
CP-ABE: High-level Design

set of attributes

$S = \{\mathsf{attr}_1, \mathsf{attr}_5\}$

$\mathsf{key} = (\mathsf{sk}_1, \mathsf{sk}_5)$

msg **secret**



s_1 ct_1

s_2 ct_2

$\vdots$ $\vdots$

s_n ct_n

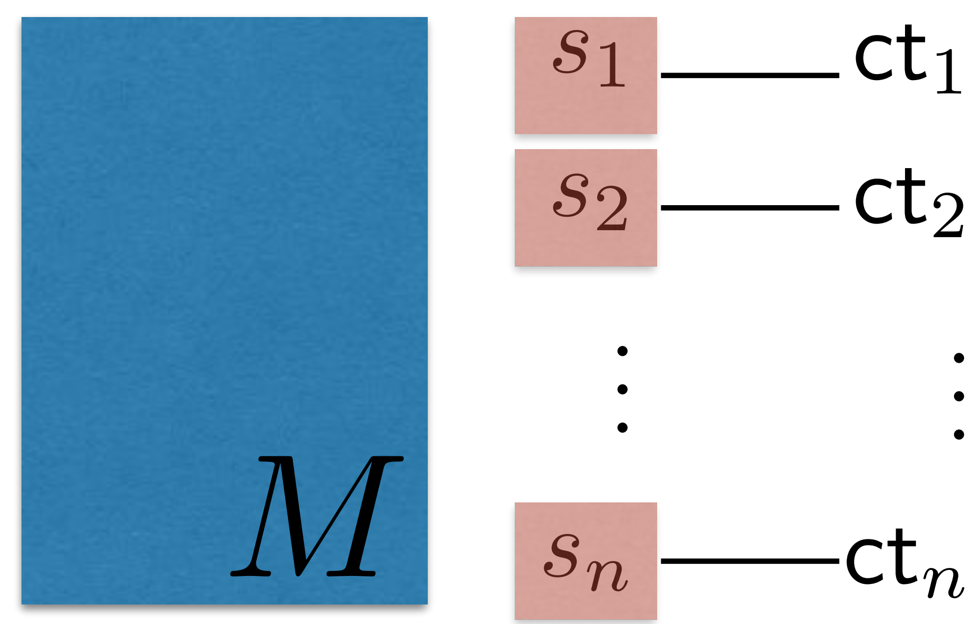
CP-ABE: High-level Design

set of attributes

$S = \{\mathsf{attr}_1, \mathsf{attr}_5\}$

$\mathsf{key} = (\mathsf{sk}_1, \mathsf{sk}_5)$

msg **secret**

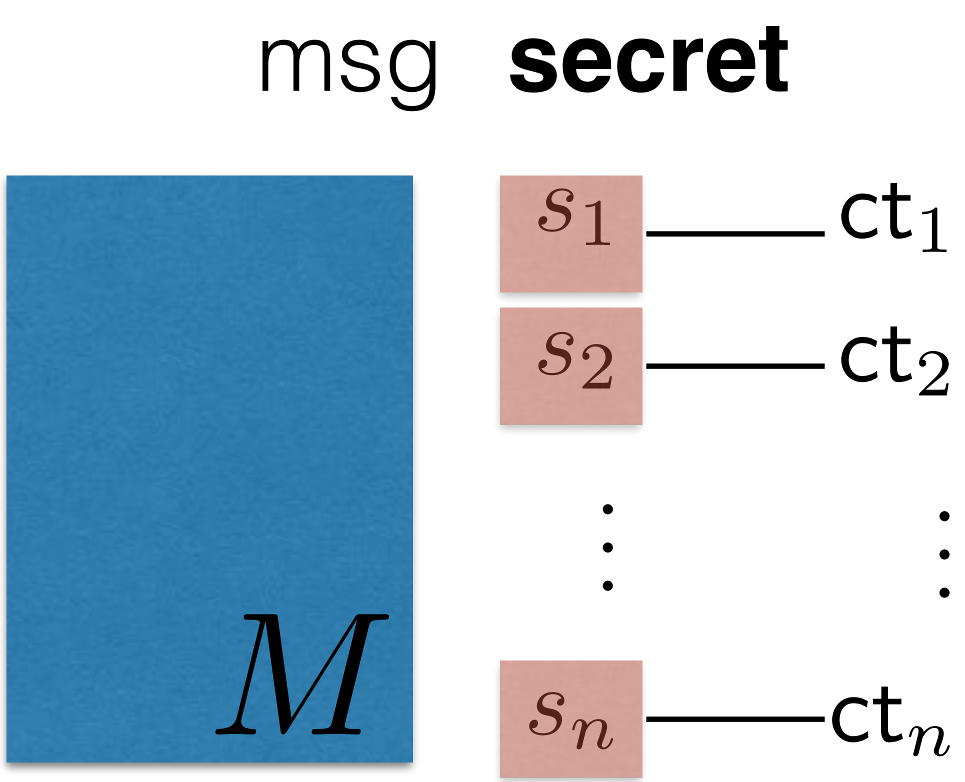


CP-ABE: High-level Design

set of attributes

$$S = \{\mathsf{attr}_1, \mathsf{attr}_5\}$$

$$\mathsf{key} = (\mathsf{sk}_1, \mathsf{sk}_5)$$



s_1

CP-ABE: High-level Design

set of attributes

$S = \{\mathsf{attr}_1, \mathsf{attr}_5\}$

$\mathsf{key} = (\mathsf{sk}_1, \mathsf{sk}_5)$

s_1

msg **secret**

M

s_1 — ct_1

s_2 — ct_2

$\vdots$ $\vdots$

s_n — ct_n

CP-ABE: High-level Design

set of attributes

$S = \{\mathsf{attr}_1, \mathsf{attr}_5\}$

$\mathsf{key} = (\mathsf{sk}_1, \mathsf{sk}_5)$

s_1 s_2

msg **secret**

M

s_1 — ct_1

s_2 — ct_2

$\vdots$ $\vdots$

s_n — ct_n

CP-ABE: High-level Design

set of attributes

$S = \{\mathsf{attr}_1, \mathsf{attr}_5\}$

$\mathsf{key} = (\mathsf{sk}_1, \mathsf{sk}_5)$

s_1 $\boxed{s_2}$

msg **secret**

M

$\boxed{s_1}$ — ct_1

$\boxed{s_2}$ — ct_2

$\vdots$ $\vdots$

$\boxed{s_n}$ — ct_n

Masking values?

CP-ABE: High-level Design

set of attributes

$S = \{\mathsf{attr}_1, \mathsf{attr}_5\}$

$\mathsf{key} = (\mathsf{sk}_1, \mathsf{sk}_5)$

s_1 s_2

msg **secret**

M

s_1 — ct_1

s_2 — ct_2

$\vdots$ $\vdots$

s_n — ct_n

Masking values?

Public key

Chen, Gay, and Wee [EC'15]

Chen, Gay, and Wee [EC'15]

- Secure under k-linear assumption; Quite fast: Type-III pairings

Chen, Gay, and Wee [EC'15]

- Secure under k-linear assumption; Quite fast: Type-III pairings
- Small universe; Restrictions on policies

Chen, Gay, and Wee [EC'15]

- Secure under k-linear assumption; Quite fast: Type-III pairings
- Small universe; Restrictions on policies
- Overcome problems *without compromising performance*
  - Perform better on most metrics

Chen, Gay, and Wee [EC'15]

Chen, Gay, and Wee [EC'15]

$$e : \mathbb{G} \times \mathbb{H} \rightarrow \mathbb{G}_T$$

Chen, Gay, and Wee [EC'15]

$$e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$$

Generators $g \in \mathbb{G}, h \in \mathbb{H}$ $\quad [a]_1 \Rightarrow g^a \quad [b]_2 \Rightarrow h^b$

Chen, Gay, and Wee [EC'15]

$$e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$$

Generators $g \in \mathbb{G}, h \in \mathbb{H}$ $\quad [a]_1 \Rightarrow g^a \quad [b]_2 \Rightarrow h^b$

Matrices $\mathbf{A}$ $\mathbf{B}$

Chen, Gay, and Wee [EC'15]

$$e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$$

Generators $g \in \mathbb{G}, h \in \mathbb{H}$ $\quad [a]_1 \Rightarrow g^a \quad [b]_2 \Rightarrow h^b$

Matrices $\mathbf{A} \quad \mathbf{B}$ $\qquad$ Vectors $\mathbf{a}^\perp \quad \mathbf{b}^\perp$

Chen, Gay, and Wee [EC'15]

$$e : \mathbb{G} \times \mathbb{H} \rightarrow \mathbb{G}_T$$

Generators $g \in \mathbb{G}, h \in \mathbb{H}$ $\quad [a]_1 \Rightarrow g^a \quad [b]_2 \Rightarrow h^b$

Matrices $\mathbf{A} \quad \mathbf{B}$ $\qquad$ Vectors $\mathbf{a}^\perp \quad \mathbf{b}^\perp$

Basis: $\boxed{([\mathbf{A}]_1, [\mathbf{b}^\perp]_1) \quad ([\mathbf{B}]_2, [\mathbf{a}^\perp]_2)}$

Chen, Gay, and Wee [EC'15]

$$e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$$

Generators $g \in \mathbb{G}, h \in \mathbb{H}$ $\quad [a]_1 \Rightarrow g^a \quad [b]_2 \Rightarrow h^b$

Matrices $\mathbf{A} \quad \mathbf{B}$ $\qquad$ Vectors $\mathbf{a}^\perp \quad \mathbf{b}^\perp$

Basis: $\boxed{([\mathbf{A}]_1, [\mathbf{b}^\perp]_1) \quad ([\mathbf{B}]_2, [\mathbf{a}^\perp]_2)}$

Each attr x: $\boxed{([\mathbf{W}_x^\top \mathbf{A}]_1, [\mathbf{W}_x^\top \mathbf{b}^\perp]_1) \quad ([\mathbf{W}_x \mathbf{B}]_2, [\mathbf{W}_x \mathbf{a}^\perp]_2)}$

Chen, Gay, and Wee [EC'15]

Basis: $\boxed{([\mathbf{A}]_1, [\mathbf{b}^\perp]_1) \qquad ([\mathbf{B}]_2, [\mathbf{a}^\perp]_2)}$

Each attr x: $\boxed{([\mathbf{W}_x^\top \mathbf{A}]_1, [\mathbf{W}_x^\top \mathbf{b}^\perp]_1) \quad ([\mathbf{W}_x \mathbf{B}]_2, [\mathbf{W}_x \mathbf{a}^\perp]_2)}$

Chen, Gay, and Wee [EC'15]

Basis: $([\mathbf{A}]_1, [\mathbf{b}^\perp]_1)$ $([\mathbf{B}]_2, [\mathbf{a}^\perp]_2)$

Each attr x: $([\mathbf{W}_x^\top \mathbf{A}]_1, [\mathbf{W}_x^\top \mathbf{b}^\perp]_1)$ $([\mathbf{W}_x \mathbf{B}]_2, [\mathbf{W}_x \mathbf{a}^\perp]_2)$

Chen, Gay, and Wee [EC'15]

Basis: $([\mathbf{A}]_1, [\mathbf{b}^\perp]_1)$ $\quad$ $([\mathbf{B}]_2, [\mathbf{a}^\perp]_2)$

Each attr x: $([\mathbf{W}_x^\top \mathbf{A}]_1, [\mathbf{W}_x^\top \mathbf{b}^\perp]_1)$ $\quad$ $([\mathbf{W}_x \mathbf{B}]_2, [\mathbf{W}_x \mathbf{a}^\perp]_2)$

$$(\mathbf{W}_x^\top \mathbf{A})^\top \mathbf{B} = \mathbf{A}^\top (\mathbf{W}_x \mathbf{B})$$

Chen, Gay, and Wee [EC'15]

Basis: $([\mathbf{A}]_1, [\mathbf{b}^\perp]_1)$ $\quad$ $([\mathbf{B}]_2, [\mathbf{a}^\perp]_2)$

Each attr x: $([\mathbf{W}_x^\top \mathbf{A}]_1, [\mathbf{W}_x^\top \mathbf{b}^\perp]_1)$ $\quad$ $([\mathbf{W}_x \mathbf{B}]_2, [\mathbf{W}_x \mathbf{a}^\perp]_2)$

$$(\mathbf{W}_x^\top \mathbf{A})^\top \mathbf{B} = \mathbf{A}^\top (\mathbf{W}_x \mathbf{B})$$

$$(\mathbf{W}_x^\top \mathbf{A})^\top \mathbf{B} \qquad \mathbf{A}^\top (\mathbf{W}_y \mathbf{B})$$

Chen, Gay, and Wee [EC'15]

Basis: $([\mathbf{A}]_1, [\mathbf{b}^\perp]_1)$ $([\mathbf{B}]_2, [\mathbf{a}^\perp]_2)$

Each attr x: $([\mathbf{W}_x^\top \mathbf{A}]_1, [\mathbf{W}_x^\top \mathbf{b}^\perp]_1)$ $([\mathbf{W}_x \mathbf{B}]_2, [\mathbf{W}_x \mathbf{a}^\perp]_2)$

$$(\mathbf{W}_x^\top \mathbf{A})^\top \mathbf{B} = \mathbf{A}^\top (\mathbf{W}_x \mathbf{B})$$

Associativity

$$(\mathbf{W}_x^\top \mathbf{A})^\top \mathbf{B} \qquad \mathbf{A}^\top (\mathbf{W}_y \mathbf{B})$$

Challenges

Small universe schemes

$[\mathbf{W}_1^\top \mathbf{A}]_1, \ldots, [\mathbf{W}_\ell^\top \mathbf{A}]_1$ in public key

Challenges

Small universe schemes

$$[\mathbf{W}_1^\mathsf{T}\mathbf{A}]_1, \ldots, [\mathbf{W}_\ell^\mathsf{T}\mathbf{A}]_1 \text{ in public key}$$

Arbitrary attributes?

Challenges

Small universe schemes

$[\mathbf{W}_1^\top \mathbf{A}]_1, \ldots, [\mathbf{W}_\ell^\top \mathbf{A}]_1$ in public key

Arbitrary attributes?

Hash function **H**

$[\mathbf{W}_x^\top \mathbf{A}]_1$ in ciphertexts

$[\mathbf{W}_x \mathbf{B}]_2$ in keys

Challenges

Hash function **H**

$[\mathbf{W}_x^\top \mathbf{A}]_1$ in ciphertexts

$[\mathbf{W}_x \mathbf{B}]_2$ in keys

Problems:

Challenges

Hash function **H**

$[\mathbf{W}_x^\top \mathbf{A}]_1$ in ciphertexts

$[\mathbf{W}_x \mathbf{B}]_2$ in keys

Problems:

- Type-III setting: $\mathbb{G}, \mathbb{H}$ different structure

Challenges

Hash function **H** $[\mathbf{W}_x^\intercal \mathbf{A}]_1$ in ciphertexts

$[\mathbf{W}_x \mathbf{B}]_2$ in keys

Problems:

- Type-III setting: $\mathbb{G}, \mathbb{H}$ different structure
- Discrete logs should be hidden

Challenges

Hash function **H** $[\mathbf{W}_x^\intercal \mathbf{A}]_1$ in ciphertexts

$[\mathbf{W}_x \mathbf{B}]_2$ in keys

Problems:

- Type-III setting: $\mathbb{G}, \mathbb{H}$ different structure
- Discrete logs should be hidden
- Use **H** to generate $[\mathbf{W}_x^\intercal \mathbf{A}]_1$

  How to generate $[\mathbf{W}_x \mathbf{B}]_2$ without explicit knowledge of $\mathbf{W}_x$

Goyal et al. KP-ABE [CCS'06]

g^{t_x} in ciphertexts $\qquad$ g^{1/t_x} in keys

Goyal et al. KP-ABE [CCS'06]

g^{t_x} in ciphertexts $\qquad$ g^{1/t_x} in keys

part of the public key

Goyal et al. KP-ABE [CCS'06]

g^{t_x} in ciphertexts

part of the public key

g^{1/t_x} in keys

master key has t_x

Goyal et al. KP-ABE [CCS'06]

g^{t_x} in ciphertexts $\qquad$ g^{1/t_x} in keys

part of the public key $\qquad$ *master key has* t_x

g^{t_x} derived directly from **H**

so that t_x is not available

Overcoming Problems

Basis: $\boxed{([\mathbf{A}]_1, [\mathbf{b}^\perp]_1) \qquad ([\mathbf{B}]_2, [\mathbf{a}^\perp]_2)}$

Each attr x: $\boxed{([\mathbf{W}_x^\top \mathbf{A}]_1, [\mathbf{W}_x^\top \mathbf{b}^\perp]_1) \qquad ([\mathbf{W}_x \mathbf{B}]_2, [\mathbf{W}_x \mathbf{a}^\perp]_2)}$

$$(\mathbf{W}_x^\top \mathbf{A})^\top \mathbf{B} = \mathbf{A}^\top (\mathbf{W}_x \mathbf{B})$$

Associativity

Overcoming Problems

Basis: $([\mathbf{A}]_1, [\mathbf{b}^{\perp}]_1)$ $([\mathbf{B}]_2, [\mathbf{a}^{\perp}]_2)$

Each attr x: $([\mathbf{W}_x^{\top}\mathbf{A}]_1, [\mathbf{W}_x^{\top}\mathbf{b}^{\perp}]_1)$ $([\mathbf{W}_x\mathbf{B}]_2, [\mathbf{W}_x\mathbf{a}^{\perp}]_2)$

$\mathbb{G}$ $\mathbb{H}$

$$(\mathbf{W}_x^{\top}\mathbf{A})^{\top}\mathbf{B} = \mathbf{A}^{\top}(\mathbf{W}_x\mathbf{B})$$

Associativity

Overcoming Problems

Basis: $([\mathbf{A}]_1, [\mathbf{b}^\perp]_1) \qquad ([\mathbf{B}]_2, [\mathbf{a}^\perp]_2)$

Each attr x: $([\mathbf{W}_x^\top \mathbf{A}]_1, [\mathbf{W}_x^\top \mathbf{b}^\perp]_1) \qquad ([\mathbf{W}_x \mathbf{B}]_2, [\mathbf{W}_x \mathbf{a}^\perp]_2)$

$$(\mathbf{W}_x^\top \mathbf{A})^\top \mathbf{B} = \mathbf{A}^\top (\mathbf{W}_x \mathbf{B})$$

Associativity

Overcoming Problems

$\mathbf{W}_x^\mathsf{T}\mathbf{A}, \mathbf{W}_x\mathbf{B}$: through $\mathbf{H}$?

Overcoming Problems

$\mathbf{W}_x^{\mathsf{T}}\mathbf{A}, \mathbf{W}_x\mathbf{B}$: through **H**?

$$\mathbf{H} \longrightarrow [\mathbf{W}_x^{\mathsf{T}}\mathbf{A}]_1$$

Overcoming Problems

$\mathbf{W}_x^{\mathsf{T}}\mathbf{A}, \mathbf{W}_x\mathbf{B}$: through **H**?

H $\longrightarrow [\mathbf{W}_x^{\mathsf{T}}\mathbf{A}]_1$

$[\mathbf{W}_x\mathbf{B}]_1$ without $\mathbf{W}_x$?

Overcoming Problems

$\mathbf{W}_x^\mathsf{T}\mathbf{A}, \mathbf{W}_x\mathbf{B}$: through **H**?

H $\longrightarrow$ $[\mathbf{W}_x^\mathsf{T}\mathbf{A}]_1$

$[\mathbf{W}_x\mathbf{B}]_1$ without $\mathbf{W}_x$?

Different approach: <u>generate keys with $[\mathbf{W}_x^\mathsf{T}\mathbf{A}]_1$, $\mathbf{B}$</u>

Overcoming Problems

$\mathbf{W}_x^\mathsf{T}\mathbf{A}, \mathbf{W}_x\mathbf{B}$: through **H**?

H $\longrightarrow [\mathbf{W}_x^\mathsf{T}\mathbf{A}]_1$

$[\mathbf{W}_x\mathbf{B}]_1$ without $\mathbf{W}_x$?

Different approach: <u>generate keys with $[\mathbf{W}_x^\mathsf{T}\mathbf{A}]_1, \mathbf{B}$</u>

Keys have different structure vs CGW

Performance Benefit

Basis: $\boxed{([\mathbf{A}]_1, [\mathbf{b}^{\perp}]_1) \qquad ([\mathbf{B}]_2, [\mathbf{a}^{\perp}]_2)}$ $\mathbb{H}$

Each attr x: $\boxed{([\mathbf{W}_x^{\top}\mathbf{A}]_1, [\mathbf{W}_x^{\top}\mathbf{b}^{\perp}]_1) \qquad ([\mathbf{W}_x\mathbf{B}]_2, [\mathbf{W}_x\mathbf{a}^{\perp}]_2)}$ $\mathbb{G}$

Performance Benefit

$\mathbb{H}$

Basis: $\boxed{([\mathbf{A}]_1, [\mathbf{b}^\perp]_1) \qquad ([\mathbf{B}]_2, [\mathbf{a}^\perp]_2)}$

$\mathbb{G}$

Each attr x: $\boxed{([\mathbf{W}_x^\top \mathbf{A}]_1, [\mathbf{W}_x^\top \mathbf{b}^\perp]_1) \qquad ([\mathbf{W}_x \mathbf{B}]_2, [\mathbf{W}_x \mathbf{a}^\perp]_2)}$

Almost all of ciphertext and key in $\mathbb{G}$

Key Features

Key Features

- Set-up time is constant.

Key Features

- Set-up time is constant.
- Ciphertexts and keys:
  - 3 elements from $\mathbb{H}$
  - 3 elements from $\mathbb{G}$ for every attribute

Key Features

- Set-up time is constant.
- Ciphertexts and keys:
  - 3 elements from $\mathbb{H}$
  - 3 elements from $\mathbb{G}$ for every attribute
- Decryption only 6 pairing operations
  - Many exponentiations, but all in $\mathbb{G}$
  - Lewko Waters' conversion

Implement, *and evaluate*

Implementation

- Python 2.7.10 using Charm 0.43 [AGMPRGR13]

Implementation

- Python 2.7.10 using Charm 0.43 [AGMPRGR13]
- MNT224 curve for pairings

Implementation

- Python 2.7.10 using Charm 0.43 [AGMPRGR13]
- MNT224 curve for pairings
- Macbook Pro laptop
  - 2.7 GHz Intel Core i5, 8GB RAM

Group Operations

(in milliseconds)

Groups	*Multiplication*	*Exponentiation*	*Hash*
$\mathbb{G}$	.009	1.266	.099
$\mathbb{H}$	.065	14.412	76.767
$\mathbb{G}_T$	.020	3.356	-

Pairing	10.243

Access Policies

$\mathsf{attr}_1 \text{ AND } \mathsf{attr}_2 \text{ AND } \ldots \text{ AND } \mathsf{attr}_n$ [GHW11]

Access Policies

$\mathsf{attr}_1 \ \mathrm{AND} \ \mathsf{attr}_2 \ \mathrm{AND} \ \ldots \ \mathrm{AND} \ \mathsf{attr}_n$ [GHW11]

$$10, 20, \ldots, 100$$

Access Policies

$\mathsf{attr}_1 \text{ AND } \mathsf{attr}_2 \text{ AND } \ldots \text{ AND } \mathsf{attr}_n$ [GHW11]

$$10, 20, \ldots, 100$$

Policies => Monotone Span Program [LW11]

Matrix has 0, 1, -1 entries

Reconstruction coefficients 0 or 1

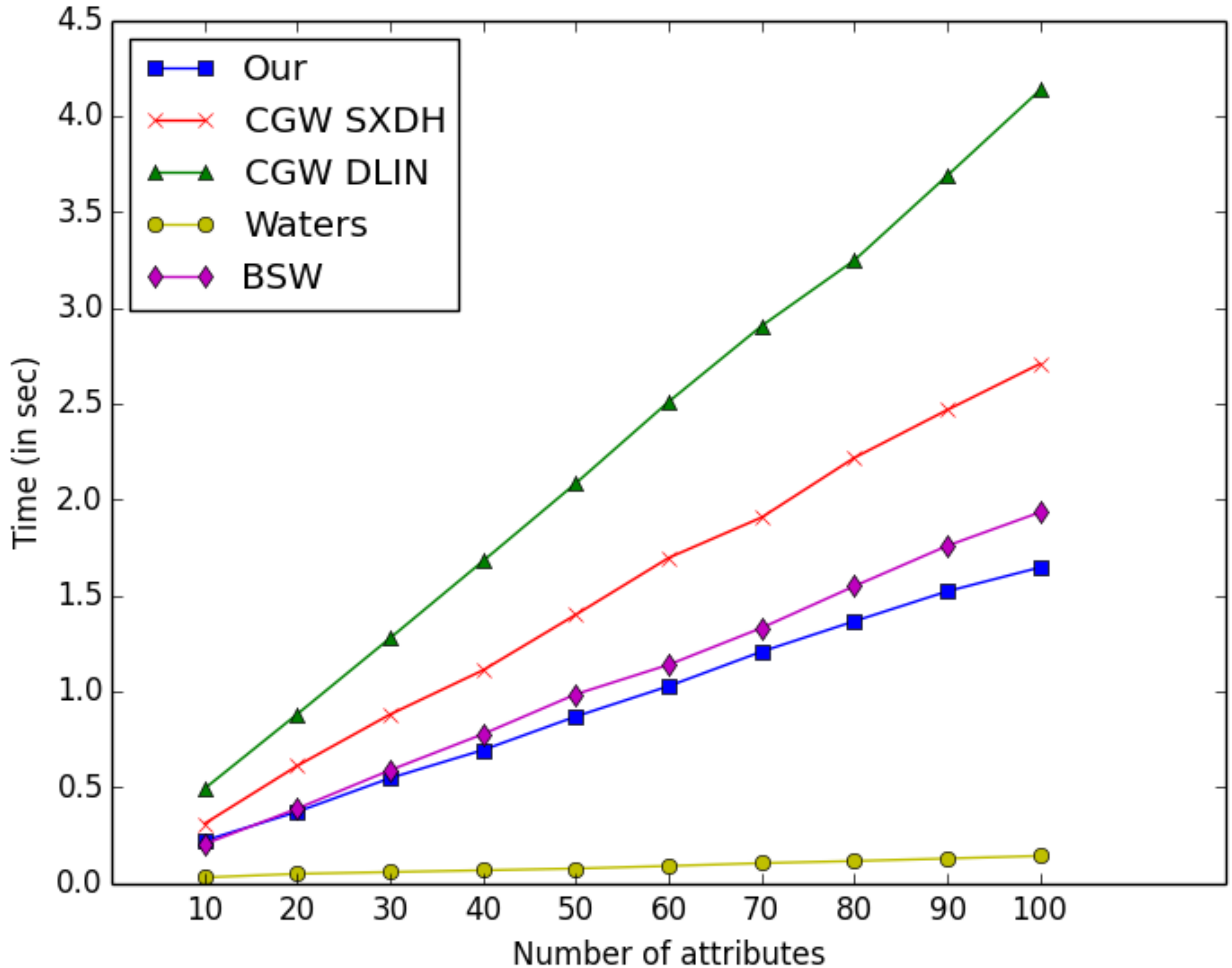
Ciphertext-Policy ABE

- Bethencourt, Sahai, and Waters (BSW) [SP'07]
- Waters [PKC'11]
- Chen, Gay, and Wee (CGW) [EC'15]
  - 1-linear (SXDH)
  - 2-linear (DLIN)

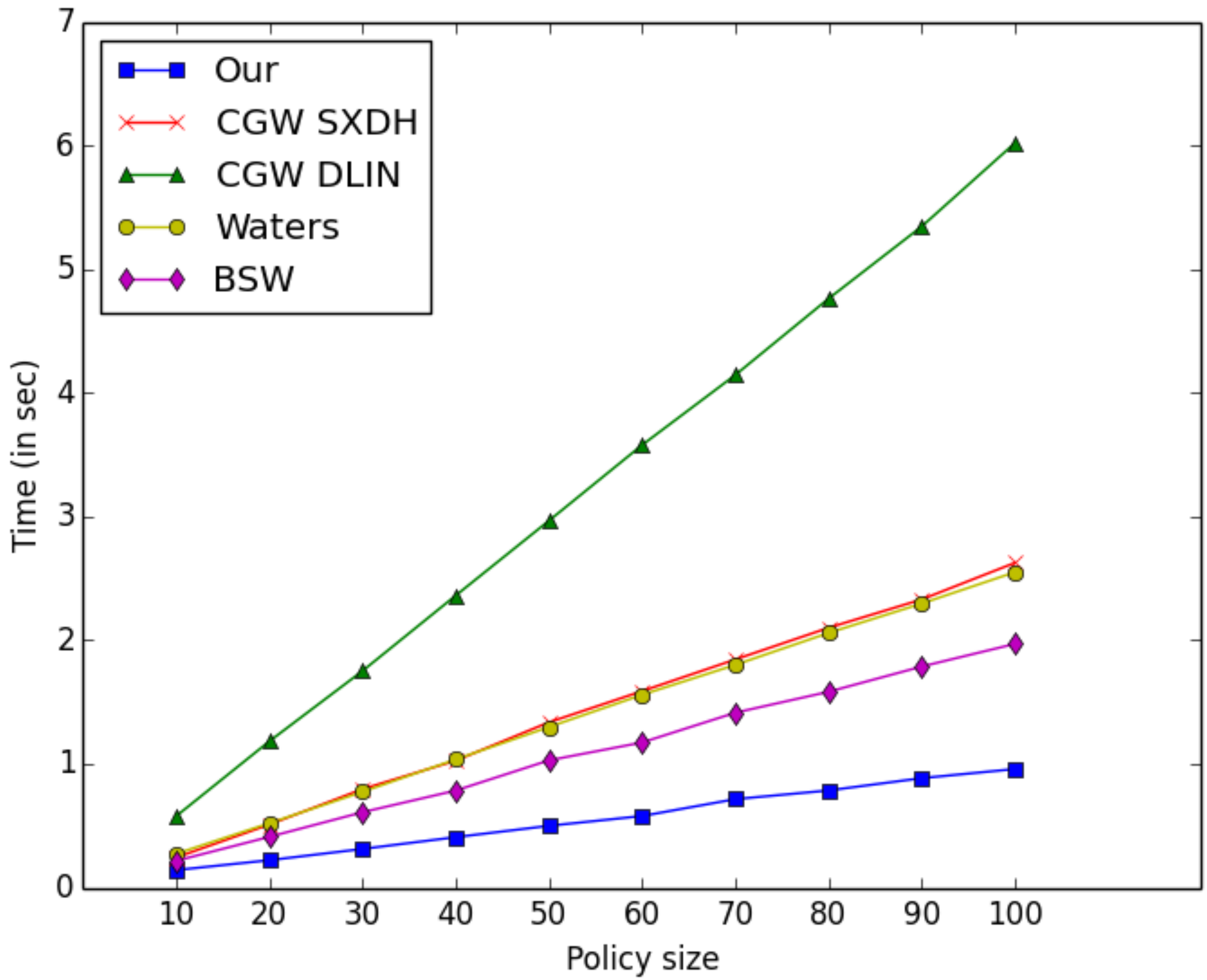
Set-up Time

Scheme	*Uni size*	*Time*
Our	-	0.11s
CGW-1	100	2.23s
CGW-2	100	5.13s
Waters	100	0.64s
BSW	-	0.08s

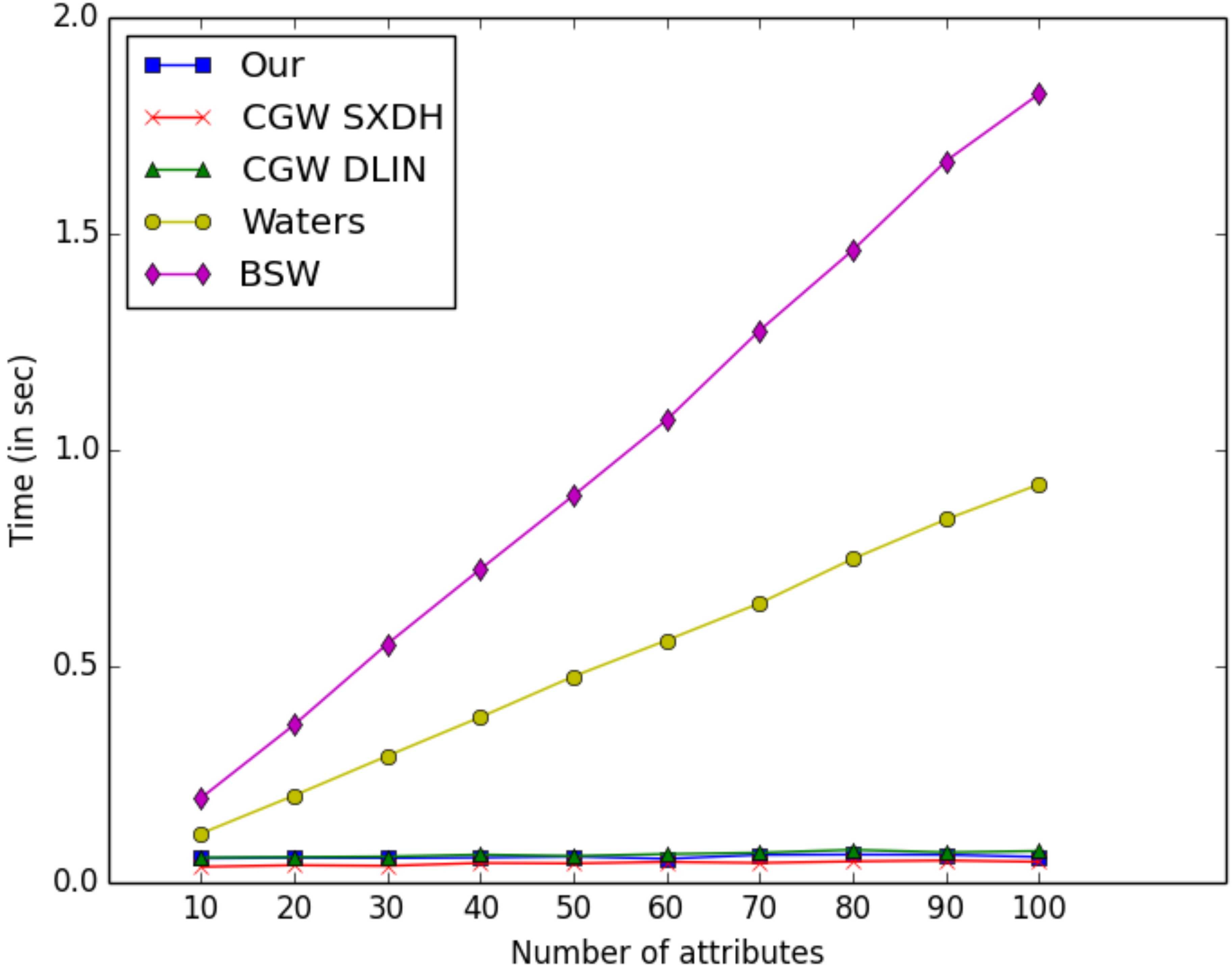
Key Generation



Encryption


Our
CGW SXDH
CGW DLIN
Waters
BSW
Time (in sec)
Policy size
0
1
2
3
4
5
6
7
10
20
30
40
50
60
70
80
90
100

Decryption



Our
CGW SXDH
CGW DLIN
Waters
BSW
Time (in sec)
2.0
1.5
1.0
0.5
0.0
10
20
30
40
50
60
70
80
90
100
Number of attributes

Conclusion

- Fast ABE schemes - good security, desirable features

Conclusion

- Fast ABE schemes - good security, desirable features
- Clean way to handle negations, multi-use of attributes

Conclusion

- Fast ABE schemes - good security, desirable features
- Clean way to handle negations, multi-use of attributes
- Optimize implementations
    - C/C++ vs Python
    - Charm's features
    - Different curve like BN

Thanks, *to you*

Paper: https://eprint.iacr.org/2017/807

Code: https://github.com/sagrawal87/ABE