GLUE: Generalizing Unbounded Attribute-Based Encryption for Flexible Efficiency Trade-Offs

Marloes Venema[1,2] Greg Alpár[3,2]

[1]University of Wuppertal, Wuppertal, Germany

[2]Radboud University, Nijmegen, the Netherlands

[3]Open Universiteit, Heerlen, the Netherlands

PKC 2023

BERGISCHE UNIVERSITÄT WUPPERTAL

Motivation

- Attribute-based encryption (ABE) is a versatile primitive that has been considered extensively to securely manage access to data
- Various properties can be supported, e.g., unlimited use of attributes, negations (NOTs)

Motivation

- Attribute-based encryption (ABE) is a versatile primitive that has been considered extensively to securely manage access to data
- Various properties can be supported, e.g., unlimited use of attributes, negations (NOTs)
- Whether properties are needed depends on the application
- Efficiency requirements may depend on the application's computational devices
- Schemes typically provide a fixed efficiency trade-off
- In particular, schemes with many (desirable) properties typically have an inefficient decryption

Motivation

- Attribute-based encryption (ABE) is a versatile primitive that has been considered extensively to securely manage access to data
- Various properties can be supported, e.g., unlimited use of attributes, negations (NOTs)
- Whether properties are needed depends on the application
- Efficiency requirements may depend on the application's computational devices
- Schemes typically provide a fixed efficiency trade-off
- In particular, schemes with many (desirable) properties typically have an inefficient decryption
- **Our goal:** creating a scheme that can support many such properties with a flexible efficiency trade-off
- Can be fine-tuned e.g., to have a very efficient decryption

High-level overview

1. Introduction to ABE
2. GLUE
3. Conclusion

High-level overview

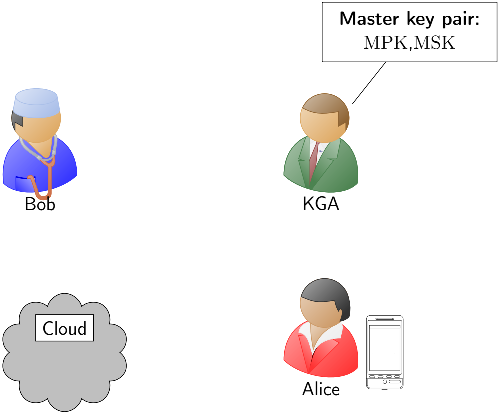
1 Introduction to ABE

2 GLUE

3 Conclusion

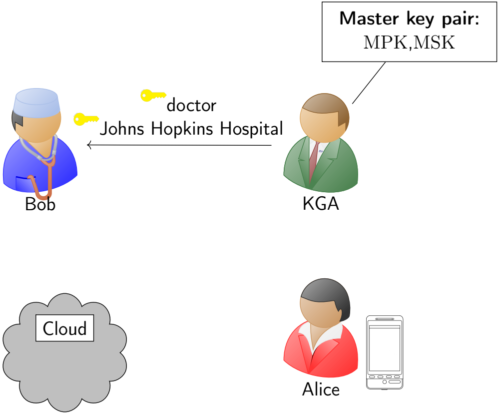
Ciphertext-policy attribute-based encryption (CP-ABE)

Setup:



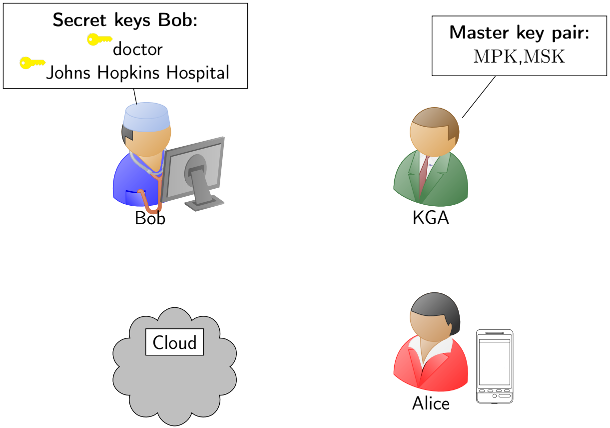
Ciphertext-policy attribute-based encryption (CP-ABE)

Key generation:



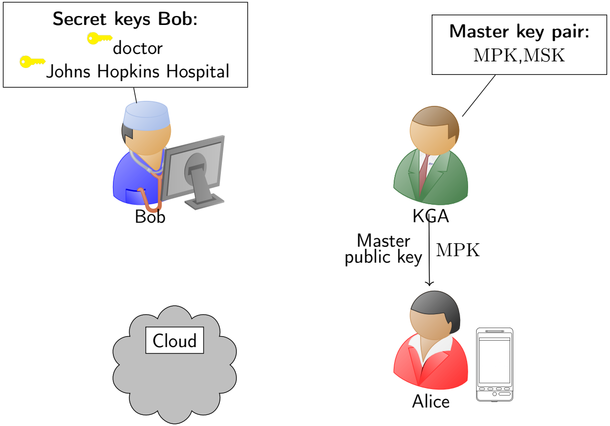
Ciphertext-policy attribute-based encryption (CP-ABE)

Key generation:



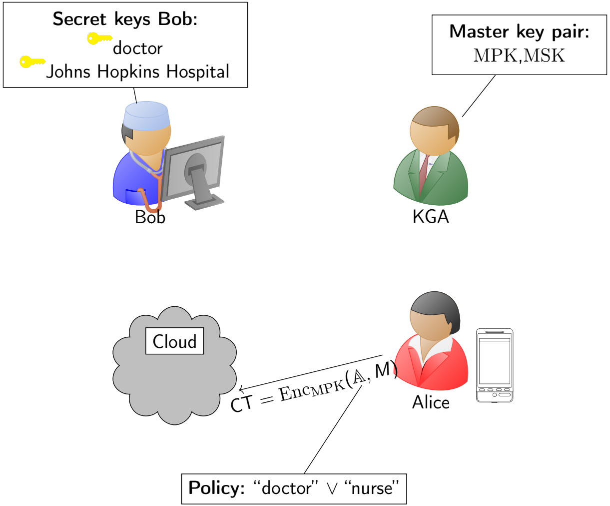
Ciphertext-policy attribute-based encryption (CP-ABE)

Encryption:

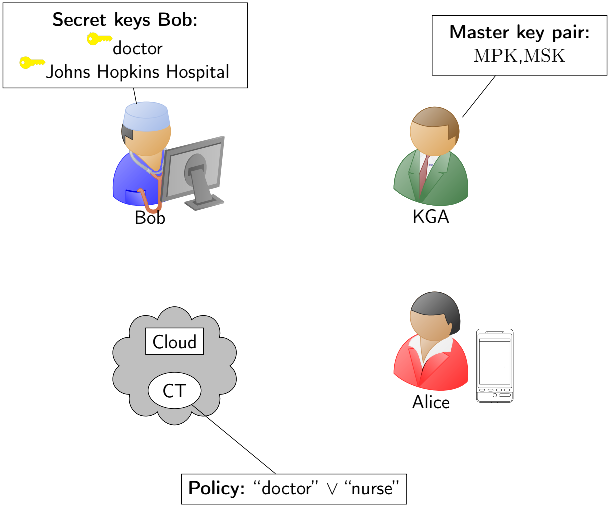


Ciphertext-policy attribute-based encryption (CP-ABE)

Encryption:

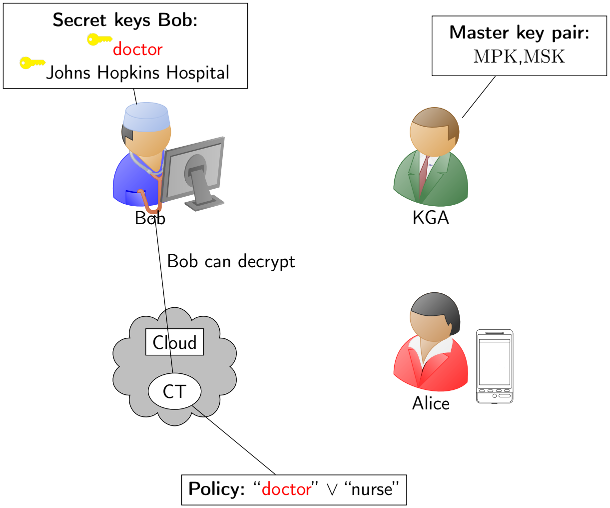


Ciphertext-policy attribute-based encryption (CP-ABE)



Ciphertext-policy attribute-based encryption (CP-ABE)

Decryption:



Enforcing access control with CP-ABE

- By its functionality, ABE implements access control
- Popular in settings in which data has to be stored on untrusted platforms

Enforcing access control with CP-ABE

- By its functionality, ABE implements access control
- Popular in settings in which data has to be stored on untrusted platforms
- The European Telecommunications Standards Institute (ETSI) considers several use cases for ABE, e.g., Cloud, IoT
- More recently, Cloudflare has presented an updated version of their Geo Key Manager: Portunus

Requirements for ABE

These use cases share many common requirements for ABE:

- **Expressive policies:** policies should support Boolean formulas consisting of AND and OR operators
- **Large universes:** attribute could be any arbitrary string, e.g., names, roles, MAC addresses
- **Unbounded:** no bounds on any parameters, such as the length of the policies or attribute sets

Requirements for ABE

These use cases share many common requirements for ABE:

- **Expressive policies:** policies should support Boolean formulas consisting of AND and OR operators
- **Large universes:** attribute could be any arbitrary string, e.g., names, roles, MAC addresses
- **Unbounded:** no bounds on any parameters, such as the length of the policies or attribute sets

Some use cases also require **non-monotonicity**, i.e., the support of negations/NOT operators in the policies. Cloudflare @ RWC 2023: "Negation is important, please include it!"

Requirements for ABE

These use cases share many common requirements for ABE:

- **Expressive policies:** policies should support Boolean formulas consisting of AND and OR operators
- **Large universes:** attribute could be any arbitrary string, e.g., names, roles, MAC addresses
- **Unbounded:** no bounds on any parameters, such as the length of the policies or attribute sets

Some use cases also require **non-monotonicity**, i.e., the support of negations/NOT operators in the policies. Cloudflare @ RWC 2023: "Negation is important, please include it!"

Storage and computational efficiency requirements may vary per use case.

Requirements for storage and computational efficiency

Examples:

- Portunus and cloud settings: fast decryption
- Internet of Things: small ciphertexts, fast encryption

Requirements for storage and computational efficiency

Examples:

- Portunus and cloud settings: fast decryption
- Internet of Things: small ciphertexts, fast encryption

Fast decryption in monotone versus non-monotone schemes

- If we do not require negations (NOTs): [AC17, RW22] have an efficient decryption
- If we do require negations, then it is more complicated

Fast decryption in monotone versus non-monotone schemes

- If we do not require negations (NOTs): [AC17, RW22] have an efficient decryption
- If we do require negations, then it is more complicated
- Most schemes supporting negations have an expensive decryption, i.e., require many pairing operations
- Some schemes support negations more efficiently, but are more restricted (i.e., bounded)

Fast decryption in monotone versus non-monotone schemes

- If we do not require negations (NOTs): [AC17, RW22] have an efficient decryption
- If we do require negations, then it is more complicated
- Most schemes supporting negations have an expensive decryption, i.e., require many pairing operations
- Some schemes support negations more efficiently, but are more restricted (i.e., bounded)
- To understand why non-monotone schemes are less efficient than monotone schemes, we observe them

Fast decryption in monotone versus non-monotone schemes

- If we do not require negations (NOTs): [AC17, RW22] have an efficient decryption
- If we do require negations, then it is more complicated
- Most schemes supporting negations have an expensive decryption, i.e., require many pairing operations
- Some schemes support negations more efficiently, but are more restricted (i.e., bounded)
- To understand why non-monotone schemes are less efficient than monotone schemes, we observe them
- At the core, all these schemes have the same underlying structure using polynomials
- Polynomials are used to support large universes

Relationship between efficiency and the polynomial's degree

- Strong relationship between the decryption efficiency and the polynomial's degree
- Roughly, higher degrees lead to faster decryption

Relationship between efficiency and the polynomial's degree

- Strong relationship between the decryption efficiency and the polynomial's degree
- Roughly, higher degrees lead to faster decryption
- For example:
  - RW13 [RW13]: 1-degree polynomial $\rightarrow$ two pairings per attribute
  - W11b [Wat08]: n-degree polynomial $\rightarrow$ two pairings per n attributes

Relationship between efficiency and the polynomial's degree

- Strong relationship between the decryption efficiency and the polynomial's degree
- Roughly, higher degrees lead to faster decryption
- For example:
  - RW13 [RW13]: 1-degree polynomial $\rightarrow$ two pairings per attribute
  - W11b [Wat08]: n-degree polynomial $\rightarrow$ two pairings per n attributes
- All unbounded schemes have a 1-degree polynomial and thus require two pairings per attribute during decryption
- High-level idea: generalize the hash!

High-level overview

1. Introduction to ABE
2. GLUE
3. Conclusion

Generalizing the polynomial-based hash

- GLUE generalizes the polynomial-based hash of RW13
- 1-degree polynomial $\rightarrow$ n-degree polynomial

Generalizing the polynomial-based hash

- GLUE generalizes the polynomial-based hash of RW13
- 1-degree polynomial $\rightarrow$ n-degree polynomial
- We also convey another parameter as a polynomial to achieve security

Generalizing the polynomial-based hash

- GLUE generalizes the polynomial-based hash of RW13
- 1-degree polynomial $\rightarrow$ n-degree polynomial
- We also convey another parameter as a polynomial to achieve security
- Note that RW13 is a monotone scheme

Partitioning to minimize pairings

GLUE generalizes the 1-degree polynomial of RW13 to an n-degree polynomial, where $n = n_k + n_c - 1$.

Partitioning to minimize pairings

GLUE generalizes the 1-degree polynomial of RW13 to an n-degree polynomial, where $n = n_k + n_c - 1$.

$\rightarrow$ partition the sets and policies in smaller subsets of maximum size n_k and n_c, respectively.

Partitioning to minimize pairings

GLUE generalizes the 1-degree polynomial of RW13 to an n-degree polynomial, where $n = n_k + n_c - 1$.

$\rightarrow$ partition the sets and policies in smaller subsets of maximum size n_k and n_c, respectively.

$\rightarrow$ number of pairings needed during decryption can be reduced by a factor in n_k and n_c, e.g., a factor of n_k if $n_k = n_c$.

Partitioning to minimize pairings

GLUE generalizes the 1-degree polynomial of RW13 to an n-degree polynomial, where $n = n_k + n_c - 1$.

$\rightarrow$ partition the sets and policies in smaller subsets of maximum size n_k and n_c, respectively.

$\rightarrow$ number of pairings needed during decryption can be reduced by a factor in n_k and n_c, e.g., a factor of n_k if $n_k = n_c$.

The higher n_k and n_c, the more efficient decryption is.

Security of GLUE and its extensions

- Security proof combines and generalizes proof techniques of [Wat08, RW13, AC16] using a new trick
- By proving security in the symbolic pair encodings framework [AC17], we achieve properties like non-monotonicity for free [AT20, Amb21]

Performance estimates

Rough estimates[1] of the storage costs of the secret keys and the ciphertexts in kilobytes (KB), where 1 KB = 1024 bytes, and the computational costs incurred by the key generation, encryption and decryption algorithms of $\mathsf{GLUE}_{(n_k,n_c)}$ and RW13, expressed in milliseconds (ms), for 10 and 100 attributes.

	Storage costs					Computational costs					
		SK		CT		KeyGen		Encrypt		Decrypt	
Scheme	$\lvert\mathrm{MPK}\rvert$	10	100	10	100	10	100	10	100	10	100
RW13	1.42	4.86	44.58	4.05	33.58	26.0	238.7	32.9	305.9	46.2	375.2
$\mathsf{GLUE}_{(3,3)}$	2.08	3.53	30.02	3.39	26.36	18.9	160.7	59.8	571.4	24.3	133.9
$\mathsf{GLUE}_{(5,5)}$	2.74	3.09	26.93	3.17	24.83	16.5	144.2	82.3	800.4	17.0	82.8
$\mathsf{GLUE}_{(10,5)}$	3.28	2.87	24.72	3.17	24.83	15.4	132.3	102.1	998.4	15.1	64.5

[1] On a 1.6 GHz Intel i5-8250U processor for the BLS12-446 curve

High-level overview

1 Introduction to ABE

2 GLUE

3 Conclusion

Conclusion

- ABE implements access control on a cryptographic level
- Various use cases require various different properties
- Previously, non-monotonicity was difficult to achieve without impacting the decryption efficiency
- GLUE addresses the need for support of negations while allowing for more efficient decryption

Thank you for your attention!

https://ia.cr/2022/613

References I

[AC16] S. Agrawal and M. Chase.
A study of pair encodings: Predicate encryption in prime order groups.
In E. Kushilevitz and T. Malkin, editors, *TCC*, volume 9563 of *LNCS*, pages 259–288. Springer, 2016.

[AC17] S. Agrawal and M. Chase.
Simplifying design and analysis of complex predicate encryption schemes.
In J.-S. Coron and J. B. Nielsen, editors, *EUROCRYPT*, volume 10210 of *LNCS*, pages 627–656. Springer, 2017.

[AHM+16] N. Attrapadung, G. Hanaoka, T. Matsumoto, T. Teruya, and S. Yamada.
Attribute based encryption with direct efficiency tradeoff.
In M. Manulis, A.-R. Sadeghi, and S. A. Schneider, editors, *ACNS*, volume 9696 of *LNCS*, pages 249–266. Springer, 2016.

[ALdP11] N. Attrapadung, B. Libert, and E. de Panafieu.
Expressive key-policy attribute-based encryption with constant-size ciphertexts.
In D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, editors, *PKC*, volume 6571 of *LNCS*, pages 90–108. Springer, 2011.

[Amb21] M. Ambrona.
Generic negation of pair encodings.
In J. A. Garay, editor, *PKC*, volume 12711 of *LNCS*, pages 120–146. Springer, 2021.

[AT20] N. Attrapadung and J. Tomida.
Unbounded dynamic predicate compositions in ABE from standard assumptions.
In *ASIACRYPT*, pages 405–436. Springer, 2020.

[Att19] N. Attrapadung.
Unbounded dynamic predicate compositions in attribute-based encryption.
In Y. Ishai and V. Rijmen, editors, *EUROCRYPT*, volume 11476 of *LNCS*, pages 34–67. Springer, 2019.

References II

[GPSW06] V. Goyal, O. Pandey, A. Sahai, and B. Waters.
Attribute-based encryption for fine-grained access control of encrypted data.
In A. Juels, R. N. Wright, and S. De Capitani di Vimercati, editors, *CCS*. ACM, 2006.

[LW11] A. B. Lewko and B. Waters.
Unbounded HIBE and attribute-based encryption.
In K. G. Paterson, editor, *EUROCRYPT*, volume 6632 of *LNCS*, pages 547–567. Springer, 2011.

[RW13] Y. Rouselakis and B. Waters.
Practical constructions and new proof methods for large universe attribute-based encryption.
In A.-R. Sadeghi, V. D. Gligor, and M. Yung, editors, *CCS*, pages 463–474. ACM, 2013.

[RW22] D. Riepel and H. Wee.
FABEO: fast attribute-based encryption with optimal security.
In H. Yin, A. Stavrou, C. Cremers, and E. Shi, editors, *CCS*, pages 2491–2504. ACM, 2022.

[VA22] M. Venema and G. Alpár.
TinyABE: Unrestricted ciphertext-policy attribute-based encryption for embedded devices and low-quality networks.
In L. Batina and J. Daemen, editors, *AFRICACRYPT*, volume 13503 of *LNCS*, pages 103–129. Springer, 2022.

[Wat08] B. Waters.
Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization.
Cryptology ePrint Archive, Report 2008/290, 2008.