

Identity-Based Encryption and Key-Policy Attribute-Based Encryption Schemes

Marco Rasori

Dept. of Information Engineering

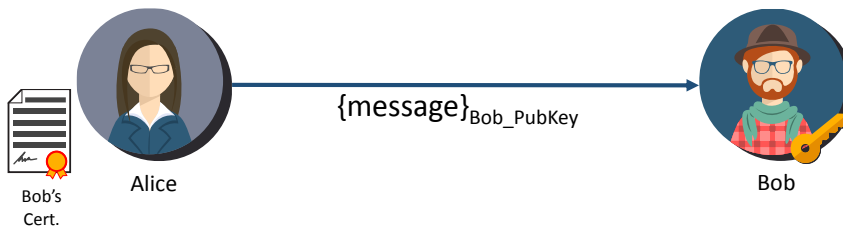
University of Pisa

marco.rasori@ing.unipi.it

2018-04-24

Public-Key (or Asymmetrical) Cryptography

- Pair of keys (*public* and *private*)
- Most used schemes: **RSA** (1977), **ECC** (1985)
- A *certificate* links a subject to a public key (certification mechanism)



Identity-Based Encryption (IBE)

- Proposed by Shamir in 1984^[1]
- The *public key* is an arbitrary string (ID in $\{0,1\}^*$), e.g., 'bob@mail.com'
- In 2001 Boneh and Franklin^[2] proposed an implementation of IBE which is based on *pairings*



[1] Shamir, Adi. "Identity-based cryptosystems and signature schemes."

[2] Boneh, Dan, and Matt Franklin. "Identity-based encryption from the Weil pairing."

IBE Algorithms (1/2)

1. **setup.** This algorithm generates the *public parameters* and a *master key*, which is kept secret.
2. **extract.** This algorithm uses the master key to generate the *private key* associated to an arbitrary public key string $ID \in \{0,1\}^*$.

Notation:

1. $(PK, MK) = \mathbf{setup}()$
2. $d_{ID} = \mathbf{extract}(MK, ID)$

PK	public parameters
MK	master key
d_{ID}	private key
ID	public key string

IBE Algorithms (2/2)

3. **encrypt.** This algorithm takes as input an arbitrary public key string $ID \in \{0,1\}^*$, the public parameters, and a message M . It returns a ciphertext C .
4. **decrypt.** It takes as input a ciphertext C , the public parameters, and a private key. It returns the message M .

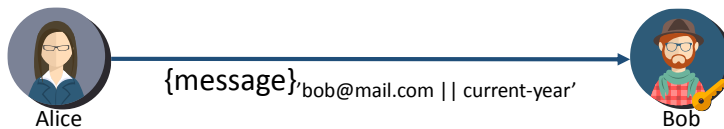
3. $C = \text{encrypt}(PK, ID, M)$
4. $M = \text{decrypt}(PK, C, d_{ID})$

Notation:

PK	public parameters
MK	master key
d_{ID}	private key
ID	public key string
M	message
C	ciphertext

Applications for IBE (1/3)

- Expiration of public keys



- 'bob@mail.com || current-year'
- Need more granularity? 'bob@mail.com || current-date'
- Alice does not need to communicate with any third party to obtain Bob's daily public key
- Alice can send messages into the future

Applications for IBE (2/3)

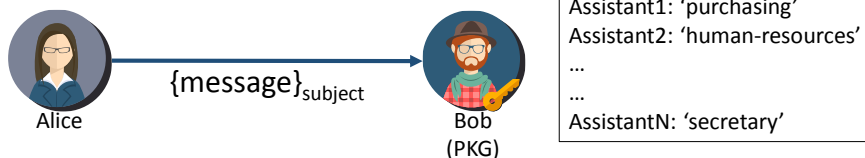
- Delegation of decryption keys
 - Bob plays the role of the PKG
 - Alice encrypts mail to Bob using the current date as IBE encryption key



- Bob goes on a trip for seven days and carries his laptop with him
- Can Bob read the emails sent by Alice during the trip? How?

Applications for IBE (3/3)

- Delegation of duties
 - Alice encrypts mail to Bob using the *subject field* as IBE encryption key
 - Bob has many assistants each responsible for a task, e.g., 'purchasing', 'human-resources', etc.
 - Bob gives a private key to each assistant depending on their responsibility




Attribute-Based Encryption

Attribute-Based Encryption^[3]

- Identity vs. Attributes


<p>$\{\text{message}\}_{\text{bob@mail.com}'}$</p>	<p>$\{\text{message}\}_{\{\text{'student', 'Pisa', 'CE'}\}}$</p>
---	---

- Ciphertexts and private keys are associated with attributes




Alice

$\{\text{message}\}_{\{\text{'student', 'Pisa'}\}}$



Bob

$\{\text{'student', 'Pisa', 'CE'}\}$



Dave

$\{\text{'professor', 'Pisa'}\}$

[3] Sahai, Amit, and Brent Waters. "Fuzzy identity-based encryption."

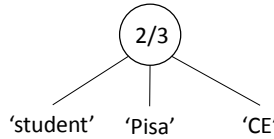
Attribute-Based Encryption



- A private key is able to decrypt a ciphertext if certain number of attributes in the ciphertext do match the ones in the private key
- Threshold gate, k -of- n



{'student', 'Pisa', 'CE'}



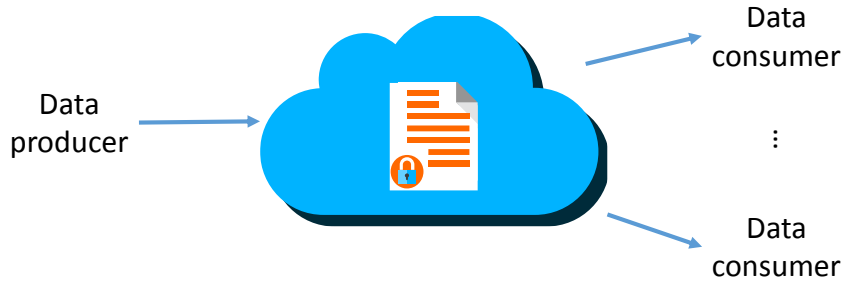
$n = 3$
 $k = 2$

The key can decrypt a ciphertext iff that ciphertext includes at least 2 of the 3 attributes of the key

{message}_{{'student', 'Pisa'}} ✓

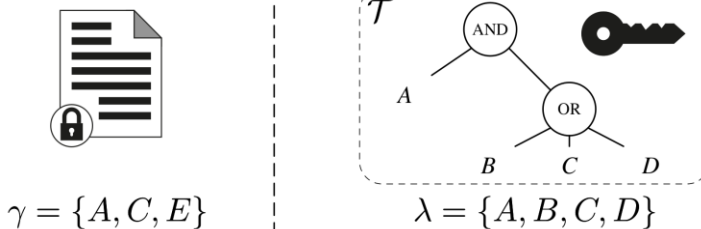
Attribute-Based Encryption

- Access control mechanism
- Untrusted storage, e.g., the cloud



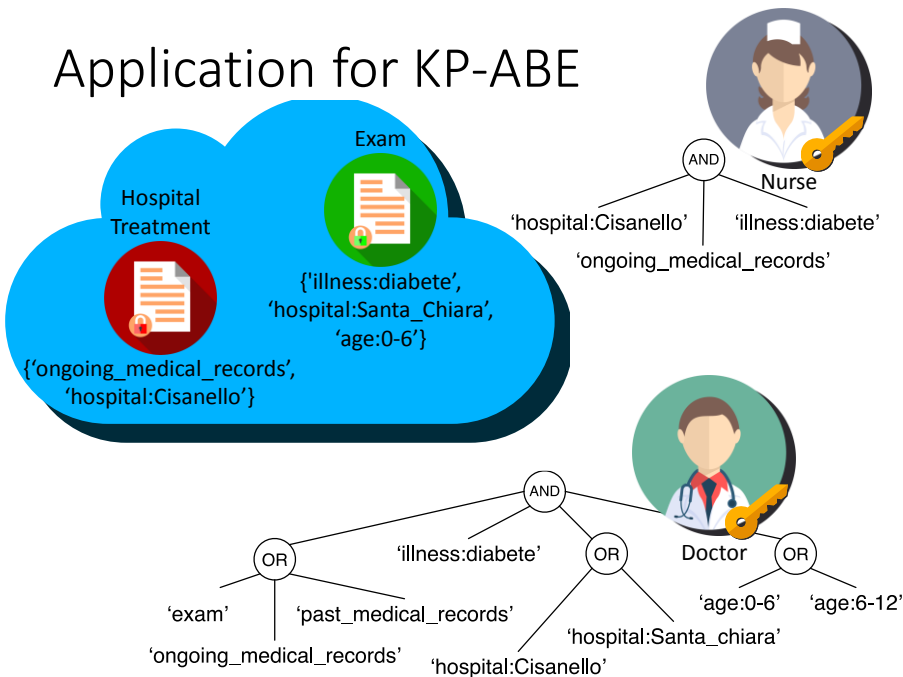
Key-Policy Attribute-Based Encryption^[4]

- A **ciphertext** is associated with an attribute set (*encryption attribute set*, γ)
- A **private key** is associated with a *policy* on the presence of some attributes (*decryption attribute set*, λ)



[4] Goyal, Vipul, et al. "Attribute-based encryption for fine-grained access control of encrypted data."

Application for KP-ABE



KP-ABE Algorithms (1/2)

1. **setup**. This algorithm generates the *public parameters* and a *master key*, which is kept secret.
2. **keygen**. This algorithm uses the master key to generate the *private key* associated to a policy T , which is defined over the decryption attribute set λ .

1. $(PK, MK) = \mathbf{setup}(U)$
2. $SK = \mathbf{keygen}(MK, T)$

Notation:

PK	public parameters
MK	master key
U	universe of attributes
SK	private key
T	policy

KP-ABE Algorithms (2/2)

3. **encrypt**. This algorithm takes as input the public parameters, an *encryption attribute set* γ , and a *message* M . It returns a *ciphertext* C .
4. **decrypt**. It takes as input a ciphertext C , and a private key SK . It returns the *message* M if the private key is able to decrypt the ciphertext, \perp otherwise.

3. $C = \mathbf{encrypt}(PK, \gamma, M)$
4. $M = \mathbf{decrypt}(C, SK)$

Notation:

PK	public parameters
SK	private key
γ	encryption attribute set
M	message
C	ciphertext

Exercise (1/3)

By using the **kpabe toolkit** installed on VictimOS, implement a KP-ABE system in which students of Computer Engineering evaluate their professors and assistants.

The software asks for the name of the course, the name of the professor/assistant to be evaluated, the evaluation (a grade from 1 to 10), and notes. Then, the software asks if the notes are confidential or non-confidential.

The software creates two ciphertexts; one containing the evaluation, and the other containing the notes. Both the ciphertexts are associated with the attributes representing the course name, the professor/assistant name, and, optionally, the attribute 'Nonconfidential', e.g., {'Cybersecurity', 'GianlucaDini', 'NonConfidential'}. The ciphertext containing the notes lacks of the attribute 'Nonconfidential' if the notes are considered confidential.

Exercise (2/3)

The subjects able to see the evaluations given by the students are the *professors*, the *assistants*, and the *head of master program*.

- The head of master program is provided with a private key which let him/her decrypt all the non-confidential ciphertexts related to all the courses of Computer Engineering.
- A professor is provided with a private key which let him/her decrypt ciphertexts related to the courses he/she teaches. Moreover, he/she can also access the non-confidential ciphertexts related to his/her assistants.
- An assistant is provided with a private key which let him/her decrypt both confidential and non-confidential ciphertexts related to him/her.

Exercise (3/3)

Write the software for the evaluation in C and use the `kpabe-enc` command within the code.

- Play the role of the TTP: setup the system and generate proper private keys for the parties.
- Play the role of the student: run the evaluation software which generates the ciphertexts.
- Play the role of the different key holders: verify that, according to the policies described above, each subject can decrypt only the portion of data he/she should be authorized to.