black hat EUROPE 2021

november 10-11, 2021

BRIEFINGS

Practical Attacks Against Attribute-based Encryption

Antonio de la Piedra (Kudelski Security Research Team) and Marloes Venema (Radboud University Nijmegen)

Introduction

Motivation

- Attribute-based encryption (ABE) is a type of public-key encryption in which the keys are associated with attributes
- Popular primitive in the enforcement of access control in IoT and cloud settings
- Several popular schemes are broken in literature
- In particular, we show that the **implementations** of the following schemes are broken:
  - YCT14
  - DAC-MACS
  - YJ14

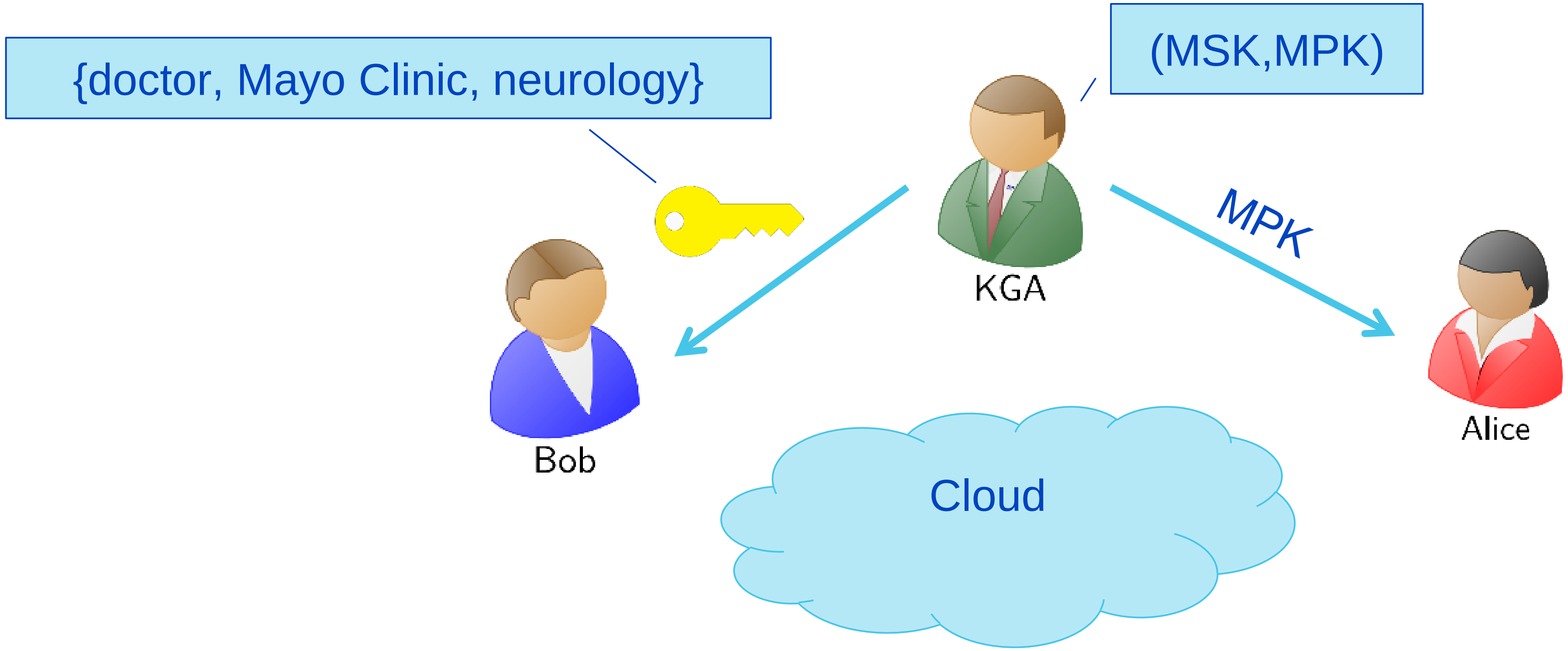
Overview

- Introduction to attribute-based encryption (ABE)
- Applications of ABE
- Closer look at the components of ABE
- How ABE schemes fail in theory
- How ABE schemes fail in practice
- Demo
- Closing remarks

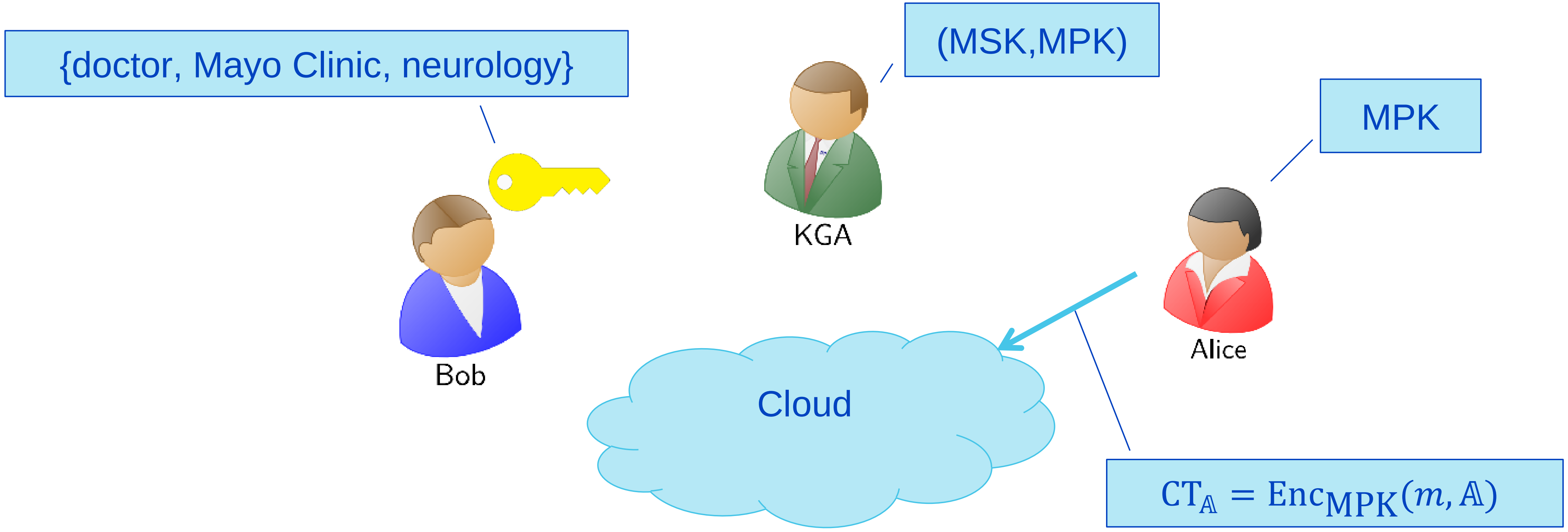
Introduction to ABE

- Two variants: key-policy (KP) and ciphertext-policy (CP) ABE
- YCT14 is a KP-ABE scheme
- DAC-MACS and YJ14 are CP-ABE schemes
- Main focus is on CP-ABE and DAC-MACS

Ciphertext-policy (CP) ABE

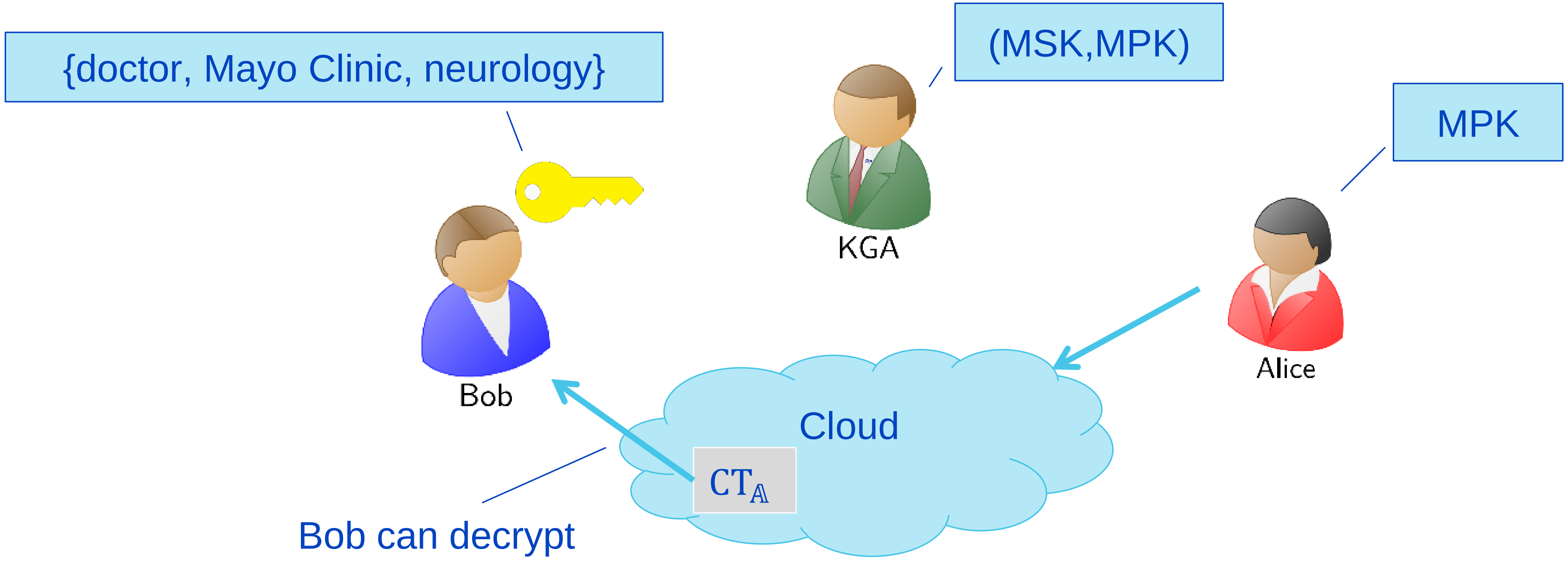


Ciphertext-policy (CP) ABE



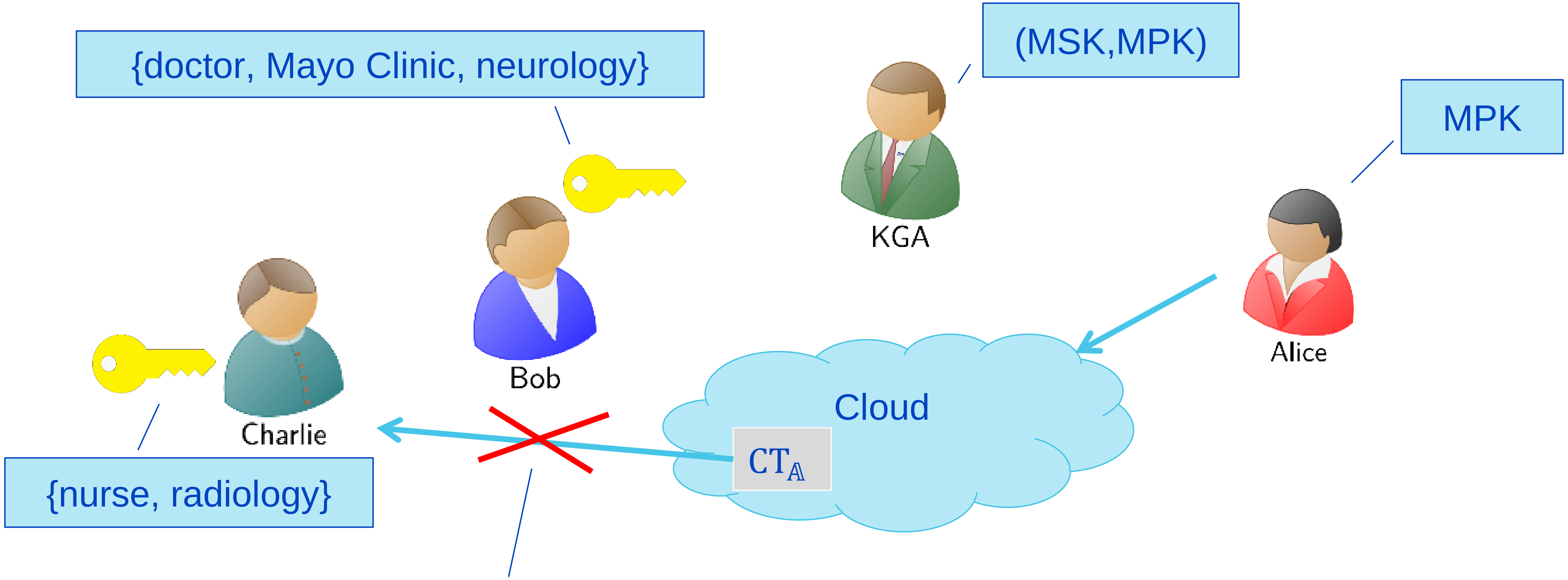
$\mathbb{A}$ is a policy: "(doctor ∨ nurse) ∧ Mayo clinic ∧ neurology"

Ciphertext-policy (CP) ABE



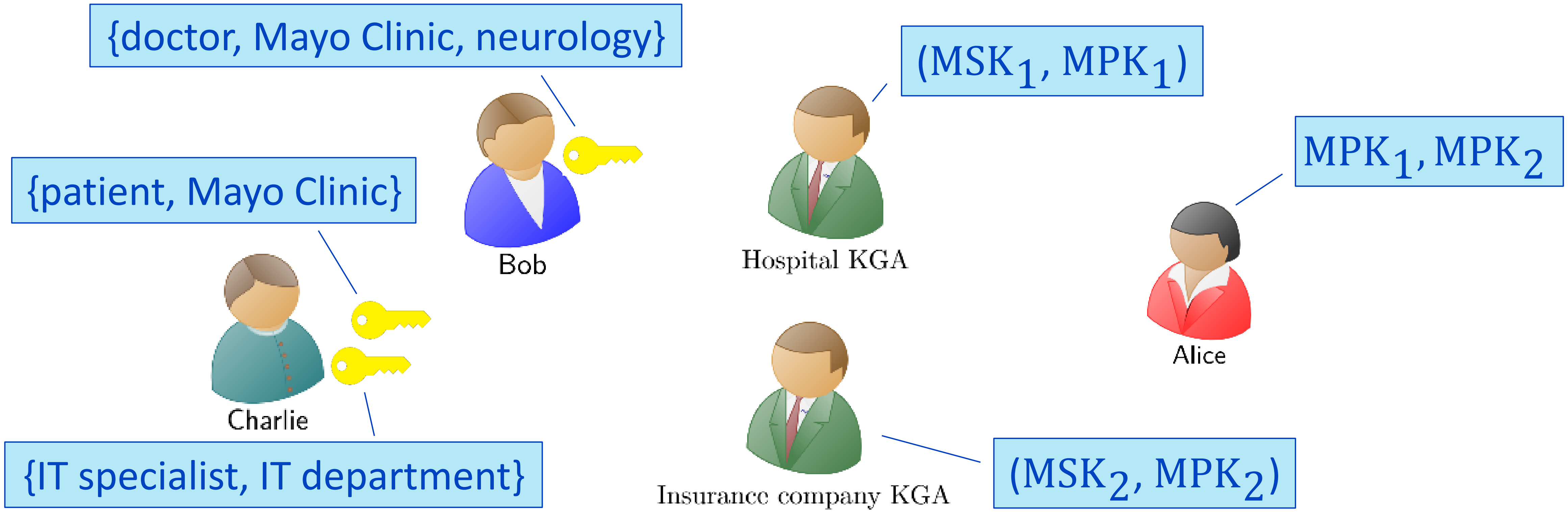
$\mathbb{A}$ is a policy: "(**doctor** ∨ nurse) ∧ **Mayo clinic** ∧ **neurology**"

Ciphertext-policy (CP) ABE



$\mathbb{A}$ is a policy: “(doctor ∨ nurse) ∧ Mayo clinic ∧ neurology”

Multi-authority ABE



Multi-authority ABE: corruption

- The previous settings are centralized: employs one central authority to generate the keys
- Multi-authority ABE allows for employment of multiple authorities
- Authorities may not trust one another
- Users may not trust all authorities
- Scheme should still be secure against corrupt authorities

Applications of ABE in the Cloud and IoT

Applications of ABE in the Cloud

- Typically utilized as an authorization mechanism in the Cloud
- Data owners e.g. Alice publish:
  - Symmetrically encrypted content e.g. media, health records, etc.
  - Encrypted keys according to a particular access policy
  - Only users e.g. Bob, Charlie, with certain attributes can decrypt

Applications of ABE in the Cloud

DAC-MACS: Effective Data Access Control for Multi-Authority Cloud Storage Systems

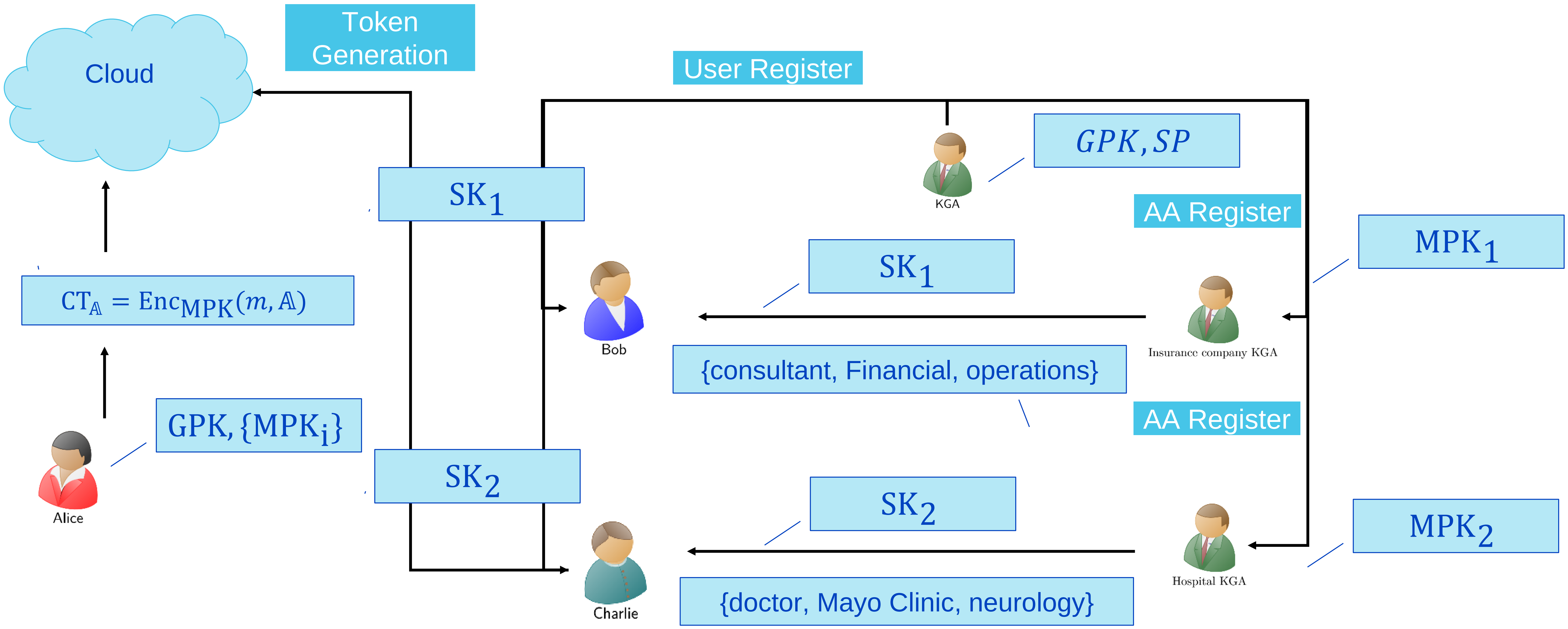
Kan Yang*, Xiaohua Jia†, Kui Ren‡
*Department of Computer Science, City University of Hong Kong, Email: kanyang3@student.cityu.edu.hk
†Department of Computer Science, City University of Hong Kong, Email: jia@cs.cityu.edu.hk
‡Department of Electrical and Computer Engineering, Illinois Institute of Technology, Email: kren@ece.iit.edu

ASCLEPIOS

DAC-MACS: Effective data access control for multiauthority cloud storage systems

K Yang, X Jia, K Ren, B Zhang… - IEEE Transactions on …, 2013 - ieeexplore.ieee.org

Data access control is an effective way to ensure data security in the cloud. However, due to data outsourcing and untrusted cloud servers, the data access control becomes a challenging issue in cloud storage systems. Existing access control schemes are no longer …



$\mathbb{A}$ is a policy: “(doctor ∨ nurse) ∧ Mayo clinic ∧ neurology”

DAC-MACS

- Master Authority 

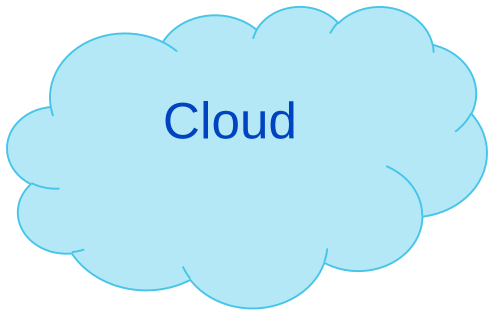
  - Generates and deploys the system
    - Registers Attribute Authorities and Users
    - Distributes master secret and public keys for each user in the system

- Attribute Authorities 

  - Independent
    - Issue and/or revoke and update user attributes
    - Manage an arbitrary number of attributes
    - Generates public attribute keys for each attribute and secret keys for each user associated to her attributes

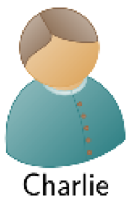
DAC-MACS

- Cloud server
  - Stores owner’s data
  - Provides access to the users
  - Generates decryption token of a ciphertext for each user using the secret keys of the user generated by the AA



- Data owners

  - Define access policies and encrypt data according the policies before storing them in the cloud
- Users



  - In order to decrypt, they send the secret key generated by a AA together with their global PK of the server and ask for a decryption token for a particular ciphertext
  - Must fulfill the associated access policy

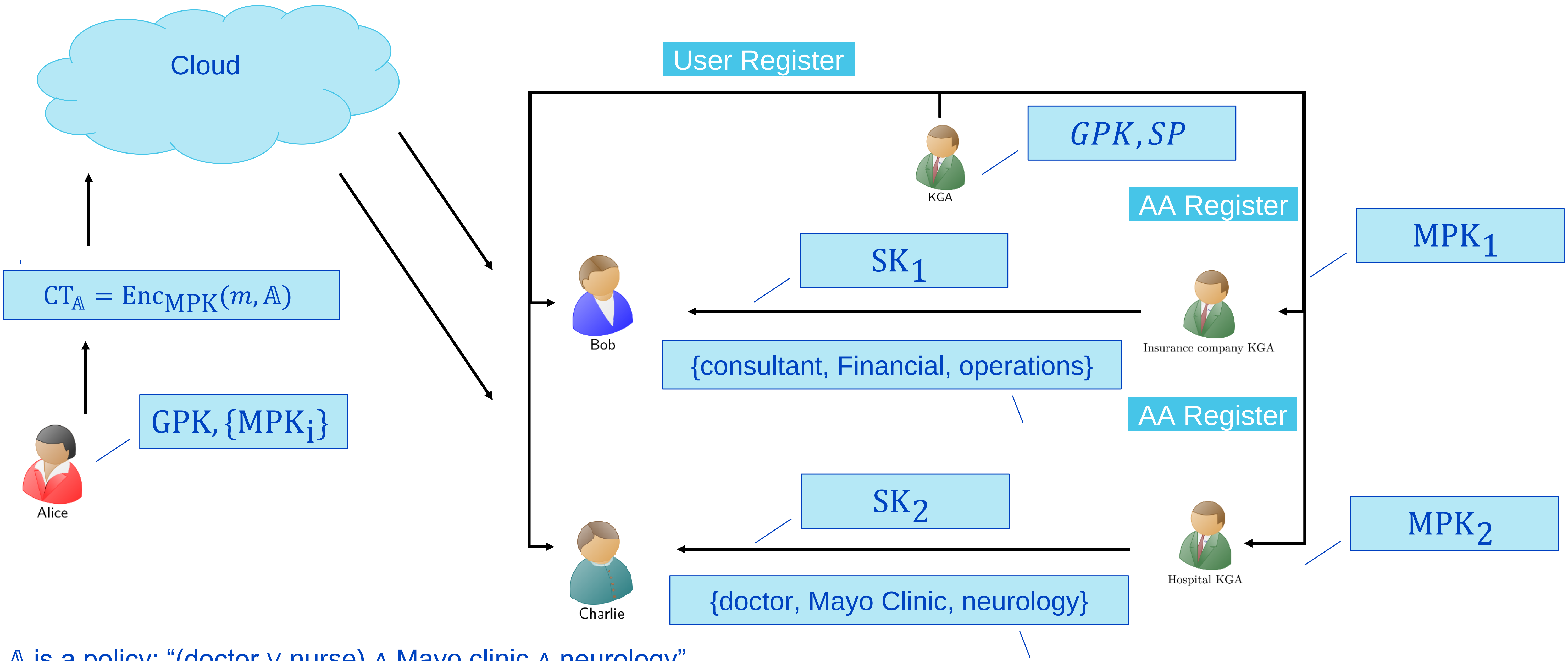
YJ14

IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 25, NO. 7, JULY 2014 1735

Expressive, Efficient, and Revocable Data Access Control for Multi-Authority Cloud Storage

Kan Yang, *Student Member, IEEE,* and Xiaohua Jia, *Fellow, IEEE*

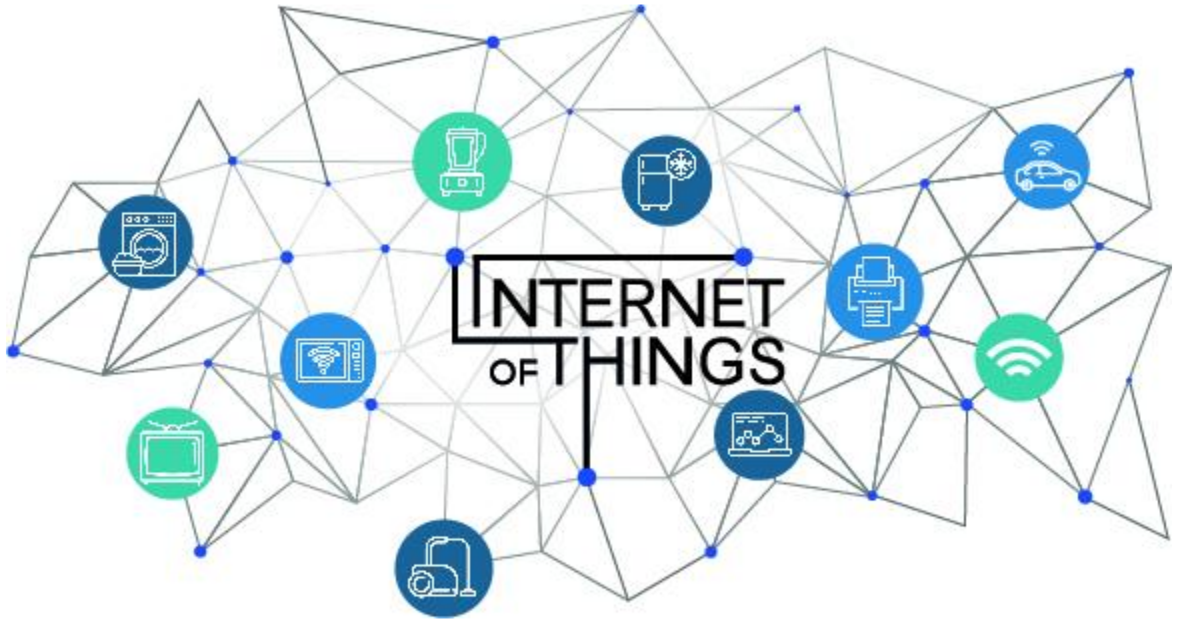
- “Improved” version of DAC-MACS
- Token generation disappears



$\mathbb{A}$ is a policy: “(doctor ∨ nurse) ∧ Mayo clinic ∧ neurology”

ABE and IoT

- Smart City
  - Data gathered from various sources and owned by different domains e.g. public and private transportation providers
  - Enforce authorization to collected data to different data owners for analysis
- Health and medical monitoring
  - Enforce access control to different parties involved e.g. nurses, doctors, patients, etc.



YCT14

- Oriented towards IoT
- Secure data transmission, storage and sharing in a distributed environment
- Single-authority and based on Elliptic-Curve Cryptography

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

ELSEVIER

A lightweight attribute-based encryption scheme for the Internet of Things

CrossMark

Xuanxia Yao [a,*], Zhi Chen [a], Ye Tian [b,c]

[a] School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, 100083, China
[b] Computer Network Information Center, Chinese Academy of Sciences, Beijing, 100190, China
[c] DNSLAB, China Internet Network Information Center, Beijing, 100190, China

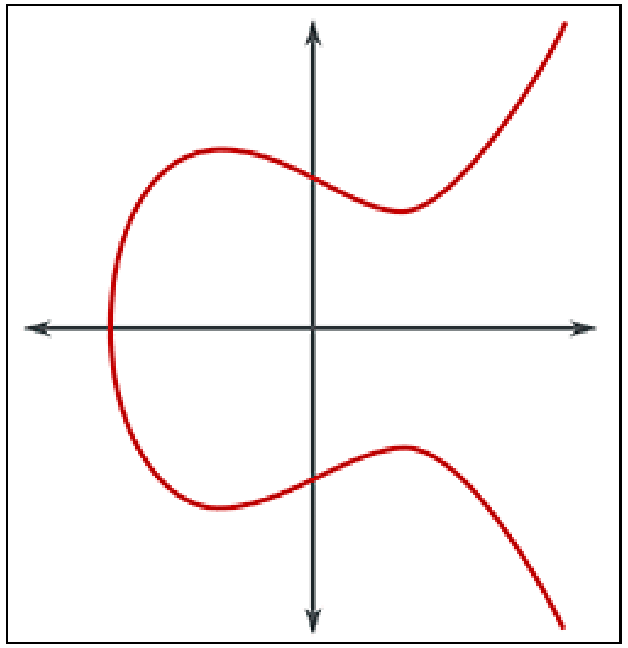
Components of ABE

Components of ABE

- Elliptic curves
- Pairings
- Secret sharing

Elliptic curves

- Groups $\mathbb{G} = \langle g \rangle$ of prime order p
- The (elliptic-curve) decisional Diffie-Hellman (DDH) is hard
- DDH: suppose g^x, g^y, Z are given, where $Z \in \mathbb{G}$ and $x, y \in \mathbb{Z}_p$, determine whether $Z = g^{xy}$



Source: Wikimedia Commons

(Elliptic curves are additive, but we often use multiplicative notation in ABE literature)

Pairings

- Efficient mapping $e: \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ such that $e(g^a, g^b) = e(g,g)^{ab}$
- Essentially allows you to exponentiate with hidden exponent
- Consider, for instance, ElGamal encryption of message M

Regular version: $PK = g^y$, $SK = y$, $CT = (A,B) = (M \cdot PK^x, g^x)$

decrypt by computing $A \cdot B^{-SK} = M$

⬇

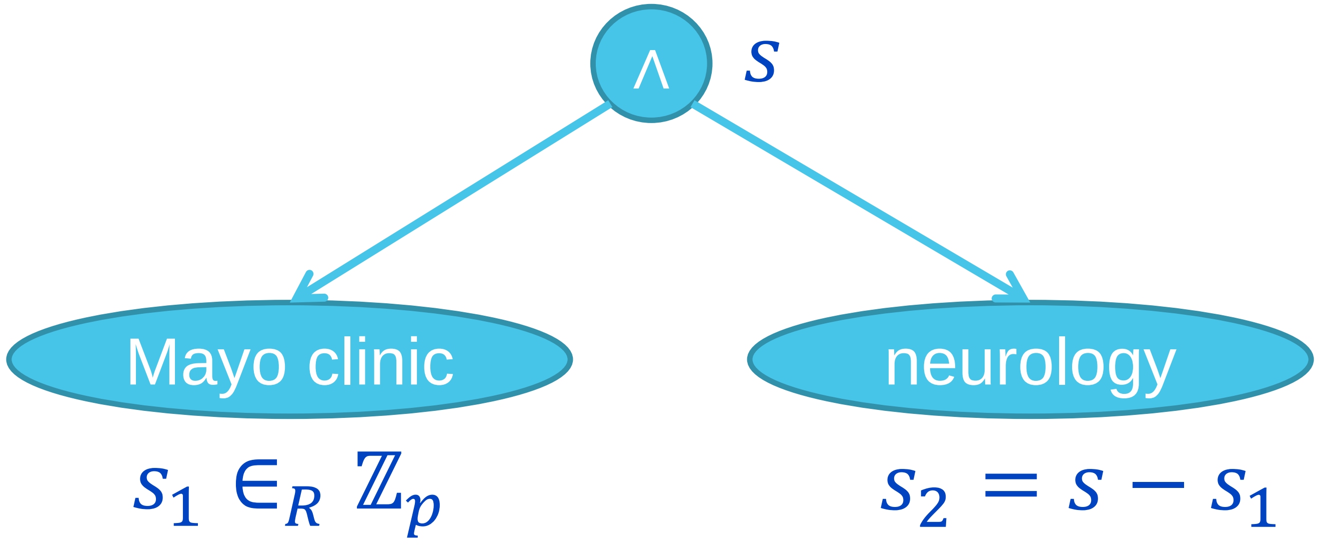
Pairing-based version: $PK = \boxed{e(g,g)^y,}$ $SK = \boxed{g^y,}$ $CT = (A,B) = (M \cdot PK^x, g^x)$

decrypt by computing $\boxed{A \cdot e(B, SK^{-1})} = M$

Secret sharing

- Computing shares of secret s with respect to some access policy
- For example, by splitting $s = s_1 + s_2$, then s_1 and s_2 are two shares corresponding to an AND gate

Example: Mayo clinic ∧ neurology



Toy example

- Global parameters: $(p, e, \mathbb{G}, \mathbb{G}_T, g)$
- Setup: $MSK = (\alpha, b, \{b_i\}_{i \in U})$, $MPK = \left(A = e(g, g)^\alpha, B = g^b, B_i = \{g^{b_i}\}_{i \in U}\right)$ where U is the universe of all attributes
- KeyGen: for a set of attributes $S = \{1,2\}$, generate $r \in_R \mathbb{Z}_p$, and output
$$SK = \left(K = g^{\alpha + rb}, K' = g^r, \{K_i = g^{rb_i}\}_{i \in S}\right)$$
- Encrypt: for policy $1 \wedge 2$, generate $s, s_1 \in_R \mathbb{Z}_p$ and $s_2 = s - s_1$, and output
$$CT = (C = M \cdot A^s, C' = g^s, C_1 = B^{s_1} B_1^s, C_2 = B^{s_2} B_2^s)$$
- Decrypt:
$$M = C \cdot e(C', K)^{-1} \cdot e(C', K_1) \cdot e(C', K_2) \cdot e(C_1, K')^{-1} \cdot e(C_2, K')^{-1}$$

How ABE schemes fail in theory

How ABE schemes fail in theory

ABE is secure if there is:

- Indistinguishability under chosen plaintext (and ciphertext) attacks
- collusion resistance: decrypting users cannot collude
- security against corruption: security with respect to honest authorities

for all ABE

for multi-authority ABE

How ABE schemes fail in theory

Proving security is difficult

attacks the ‘lightweight’ scheme by Yao, Chen and Tian (2014)

Many schemes turn out to be broken, as shown by

- Tan, Yeow and Hwang (IEEE Internet of Things journal, 2019)
- Herranz (IEEE Access journal, 2020)
- Venema and Alpár (CT-RSA, 2021)

proves that many ‘pairing-free’ schemes are broken

formalize a methodology to find attacks on pairing-based ABE schemes
\+ add eleven attacks on existing schemes, including DAC-MACS and its “improvement”

Pairing-free elliptic-curve schemes

Herranz (IEEE Access journal, 2020):

- Herranz showed that several pairing-free elliptic-curve schemes are broken
- To illustrate to the practical community that these are really broken
- Generally known in the theoretical community that secure ABE based on e.g. DDH only does not exist

Main problem: pairing-free schemes take what's in the exponent of pairing-based schemes out of the exponent and just do everything in the integer space
→ gives attackers much more power → enables attacks
This is also the problem of YCT14

Venema-Alpár framework

Venema and Alpár (CT-RSA, 2021):

- Construct a framework for analyzing pairing-based schemes
- Considers a shorter notation based on pair encodings
- Consists of stronger attack models that simplify the analysis
- For example, decryption attack:
  attacker decrypts a ciphertext even though he does not have an authorized key
- Complete decryption attack: attacker can decrypt any ciphertext
- Two of the broken schemes presented are DAC-MACS and YJ14

Attack on DAC-MACS

- Global parameters: $\left(p, e, \mathbb{G}, \mathbb{G}_T, g, g^b\right)$
- AuthoritySetup i: $MSK_i = (\alpha_i, b_i)$, $MPK_i = \left(e(g,g)^{\alpha_i}, g^{\frac{1}{b_i}}\right)$
- KeyGen: $\mathrm{SK} = (\boxed{x_1, x_2,} g^{x_2}), SK_i' = \left(g^{\frac{\alpha_i}{x_1} + x_2 b + \frac{r_i b}{b_i}}, g^{\frac{r_i b}{b_i}}, g^{r_i b}, \ldots\right)$ where $r_i \in_R \mathbb{Z}_p$
- Encrypt: $CT = (M \cdot (\prod_i e(g,g)^{\alpha_i})^s, g^s, g^{\frac{s}{b_i}}, \ldots)$ where $s \in_R \mathbb{Z}_p$

Attack: $M \cdot (\prod_i e(g,g)^{\alpha_i})^s \cdot e\left(g^{\frac{\alpha_i}{x_1} + x_2 b + \frac{r_i b}{b_i}}, g^s\right)^{-x_1} \cdot e(g^b, g^s)^{x_1 x_2} \cdot e\left(g^{r_i b}, g^{\frac{s}{b_i}}\right)^{x_1} = M$

Main issue is knowledge of x_2

Attack on YJ14

- As mentioned, this scheme is similar to DAC-MACS
- One of the “improvements”: x_2 is encrypted
- However, authorities can decrypt it
- Corruption still leads to the attack

How ABE schemes fail in practice

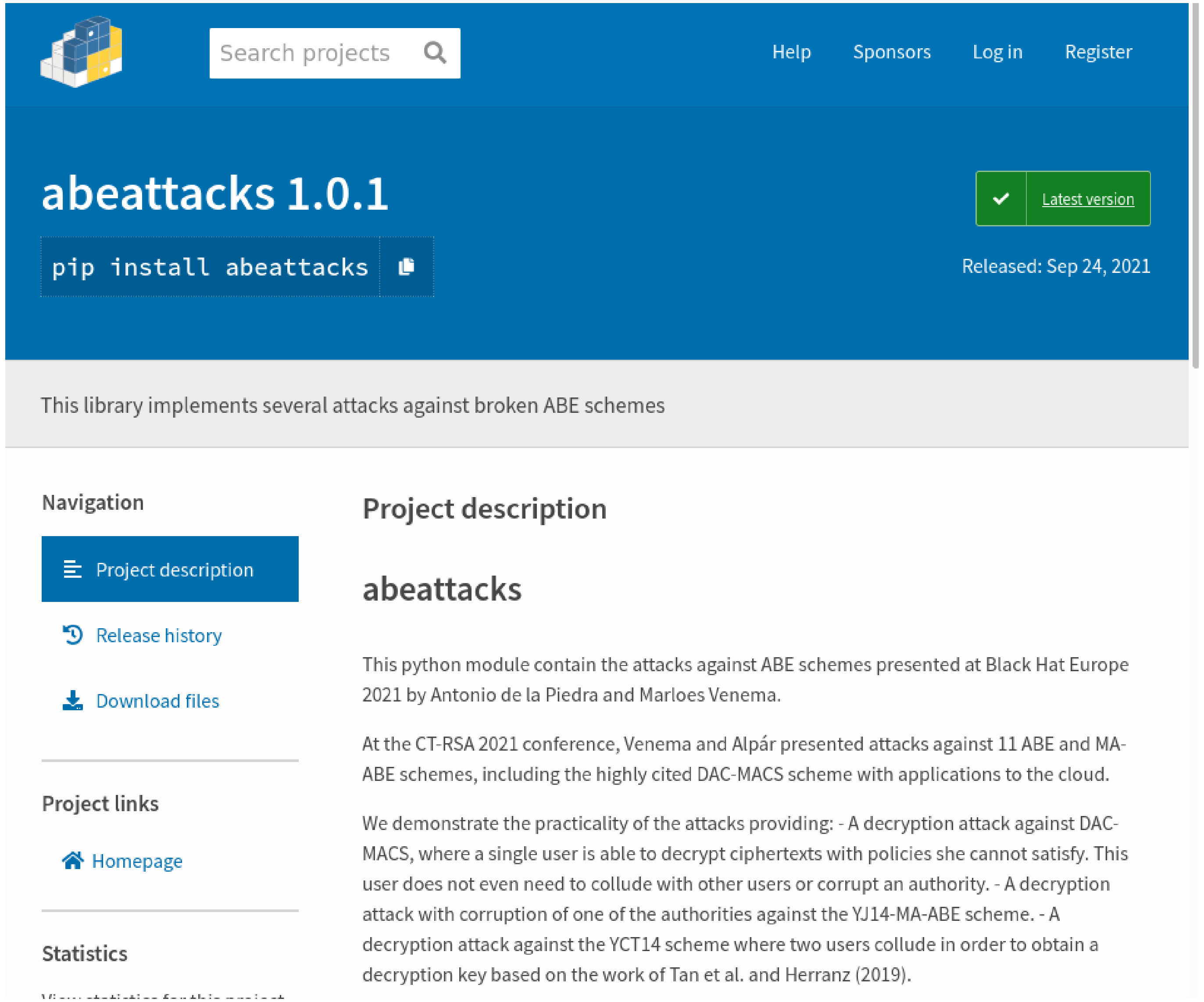
Open-source ABE implementations

- Fentec Project GoFE: https://github.com/fentec-project/gofe
- Zeutro OpenABE: https://github.com/zeutro/openabe
- CHARM Framework: https://github.com/JHUISI/charm
  - Provides DAC-MACS, YJ14 and YCT14
- Fraunhofer AISEC RABE: https://github.com/Fraunhofer-AISEC/rabe
  - Provides YCT14

CVE-2021-37587 and CVE-2021-37588

- In CHARM any single user can decrypt DAC-MACS data, even if she cannot fulfill the access policy.
  - Complete decryption attack
- In YJ14 one only needs to corrupt one of the authorities to be able to decrypt any ciphertext.
  - Complete decryption attack with corruption of one of the authorities
- Any two users can collude to achieve the ability to decrypt YCT14 data encrypted with an access policy that they cannot fulfill

Learning more about the attacks



- Python module implementing the attacks
  - https://pypi.org/project/abeattacks/
- Jupyter notebooks describing the attacks step by step
  - https://github.com/kudelskisecurity/abeattacks/jupyter
- Docker image with everything ready to use
  - https://github.com/kudelskisecurity/abeattacks/docker

Demo

Concluding remarks

- ABE is a popular primitive
- Many schemes have been broken in literature
- Some of these schemes have been implemented
- We have given practical attacks on these implementations
- These implementations should thus not be used

Main lesson: be very careful with schemes that have integer exponents in the keys
Use schemes that do not do this, e.g.
- LW11, RW15 in the multi-authority setting
- FAME in the single-authority setting

Thank you for your attention!

Questions?