Attribute-Based Encryption, Variants and Pairing-based Instantiations

MELISSA CHASE (MICROSOFT RESEARCH)

Overview

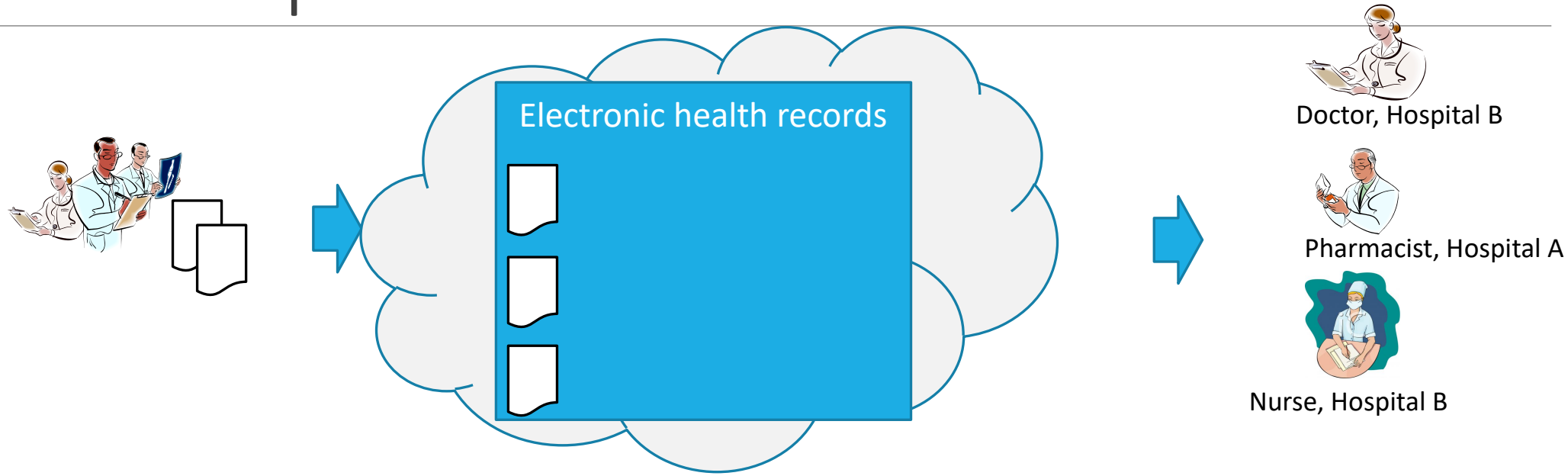
What is Attribute Based Encryption (ABE)?

Applications

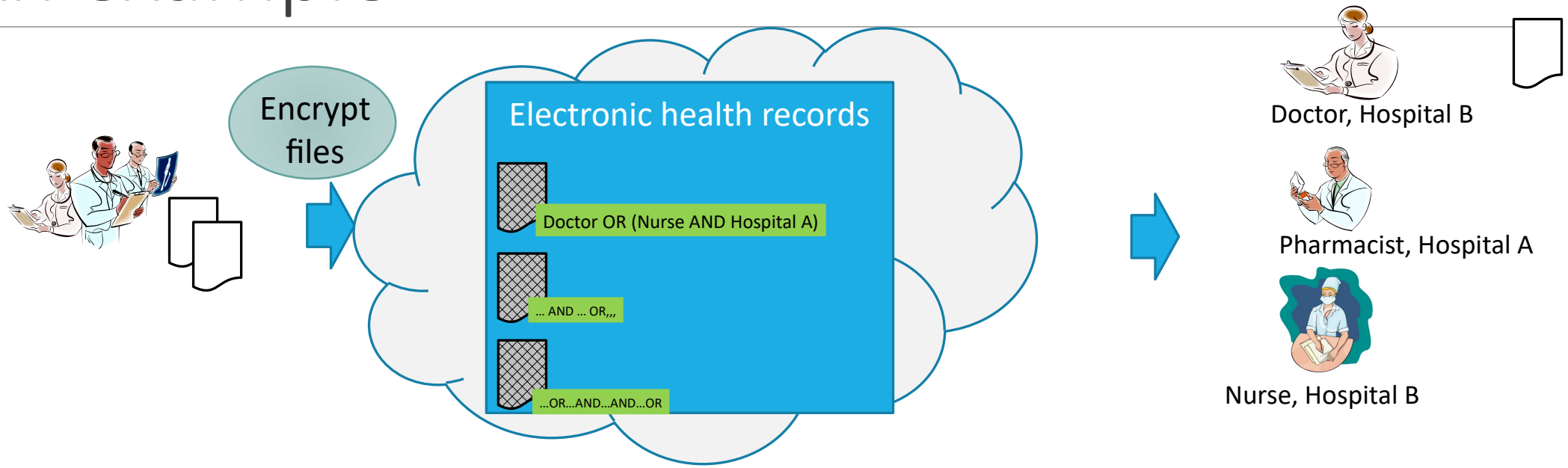
Variants

Constructing ABE from pairings

An example

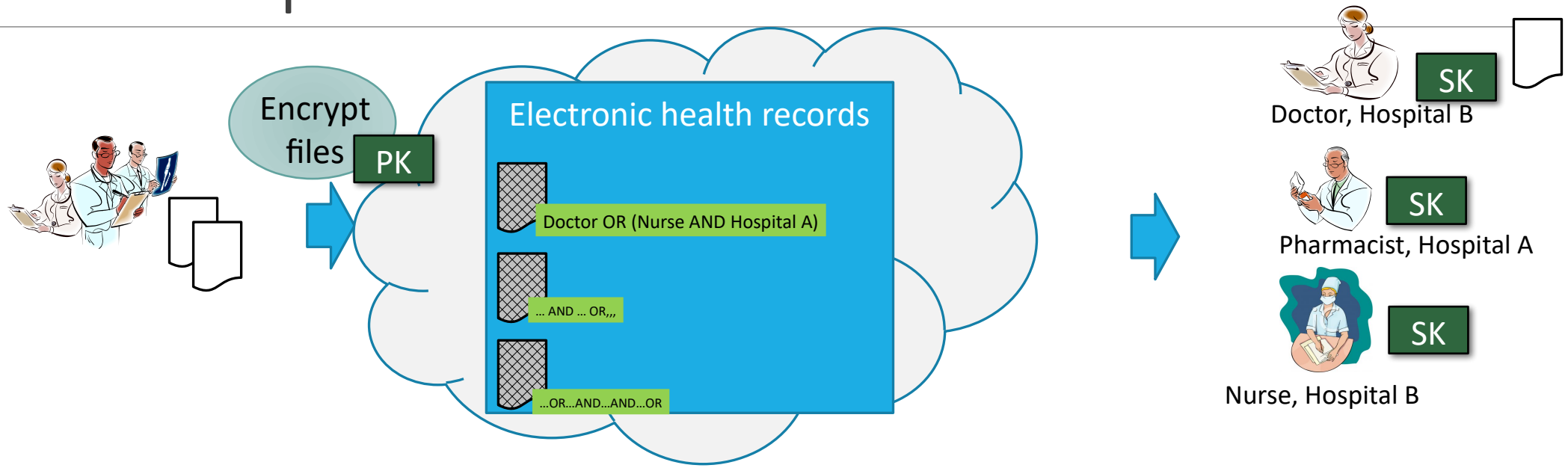

Electronic health records
Doctor, Hospital B
Pharmacist, Hospital A
Nurse, Hospital B

An example



Can we build this from public key encryption?

An example

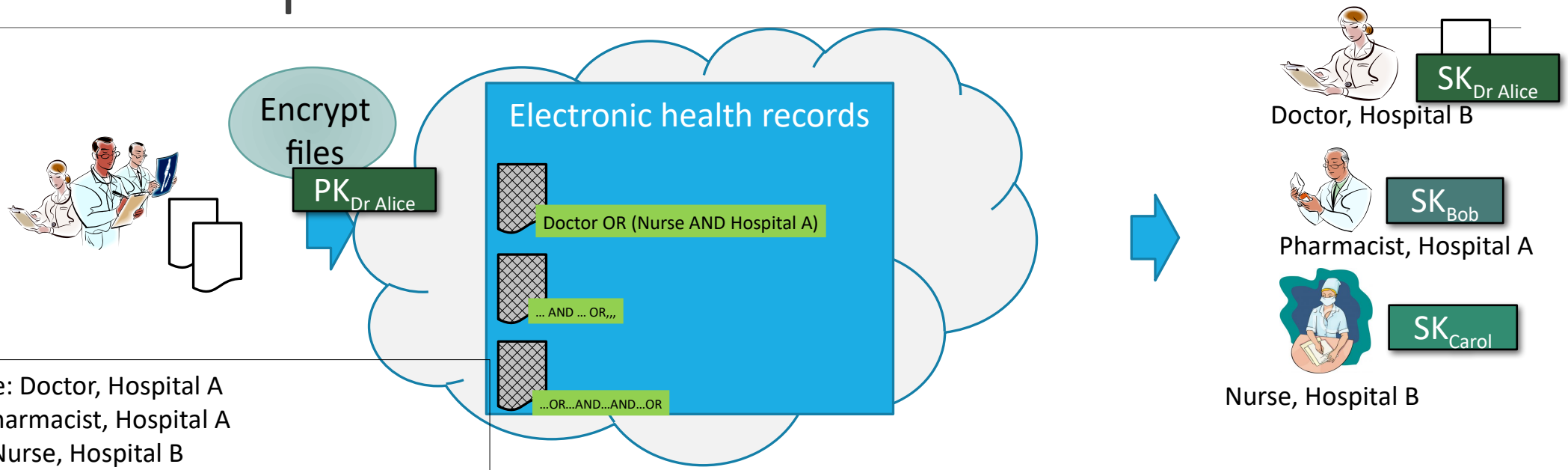


Can we build this from public key encryption?

Solution 1: Encrypt flies under master public key, give all recipients secret key

- Every recipient has access to everything

An example



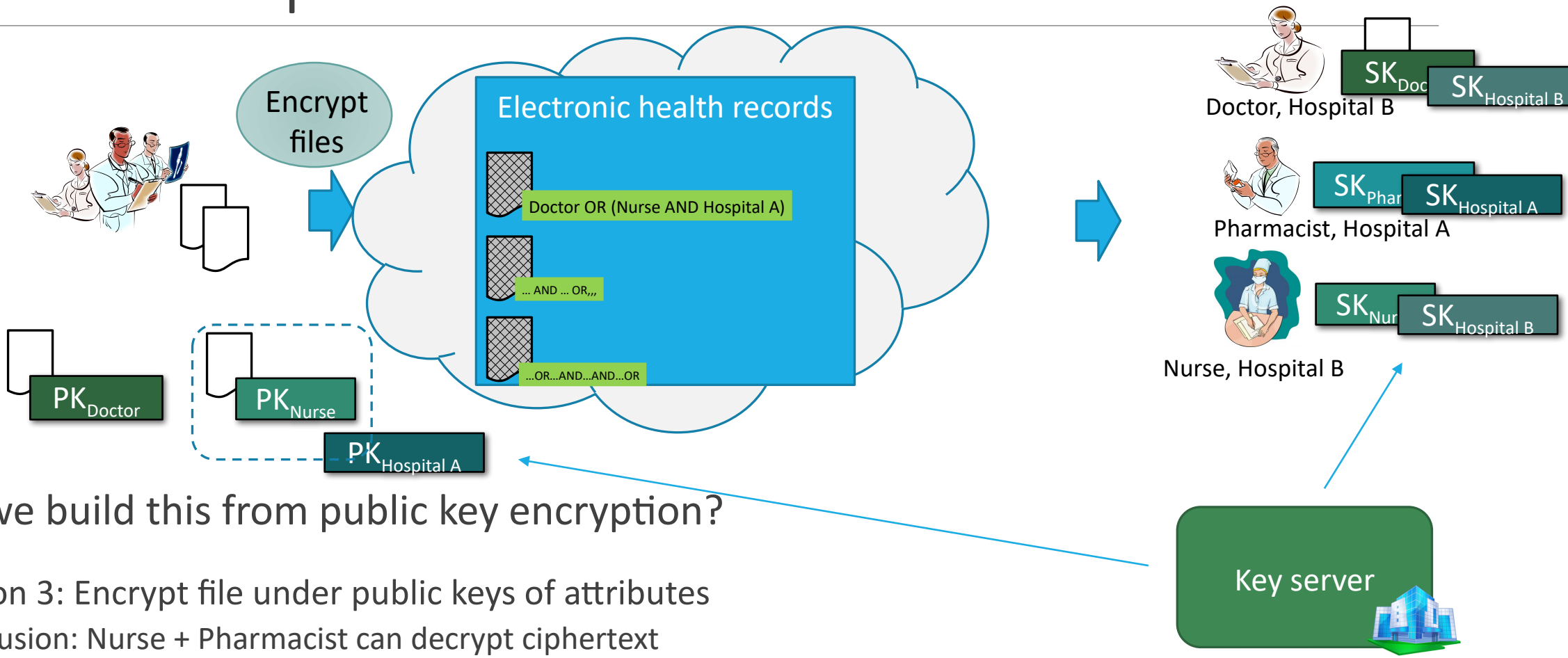
- PKDr Alice: Doctor, Hospital A
- PKBob: Pharmacist, Hospital A
- PKCarol: Nurse, Hospital B
- …

Can we build this from public key encryption?

Solution 2: Encrypt file under public key of each desired recipient

- Must know public keys and attributes of all recipients
- Separate encryption for each recipient

An example

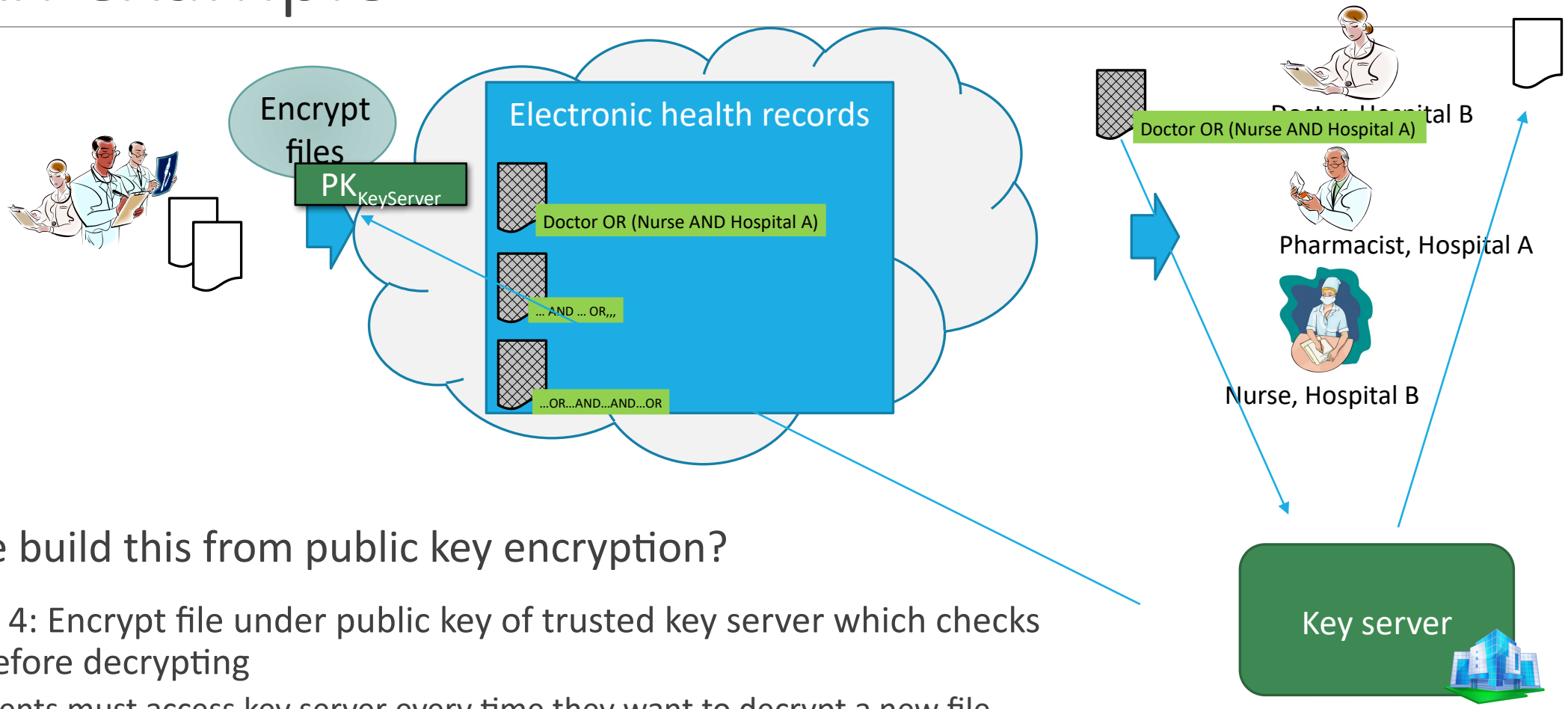


Can we build this from public key encryption?

Solution 3: Encrypt file under public keys of attributes

- Collusion: Nurse + Pharmacist can decrypt ciphertext

An example

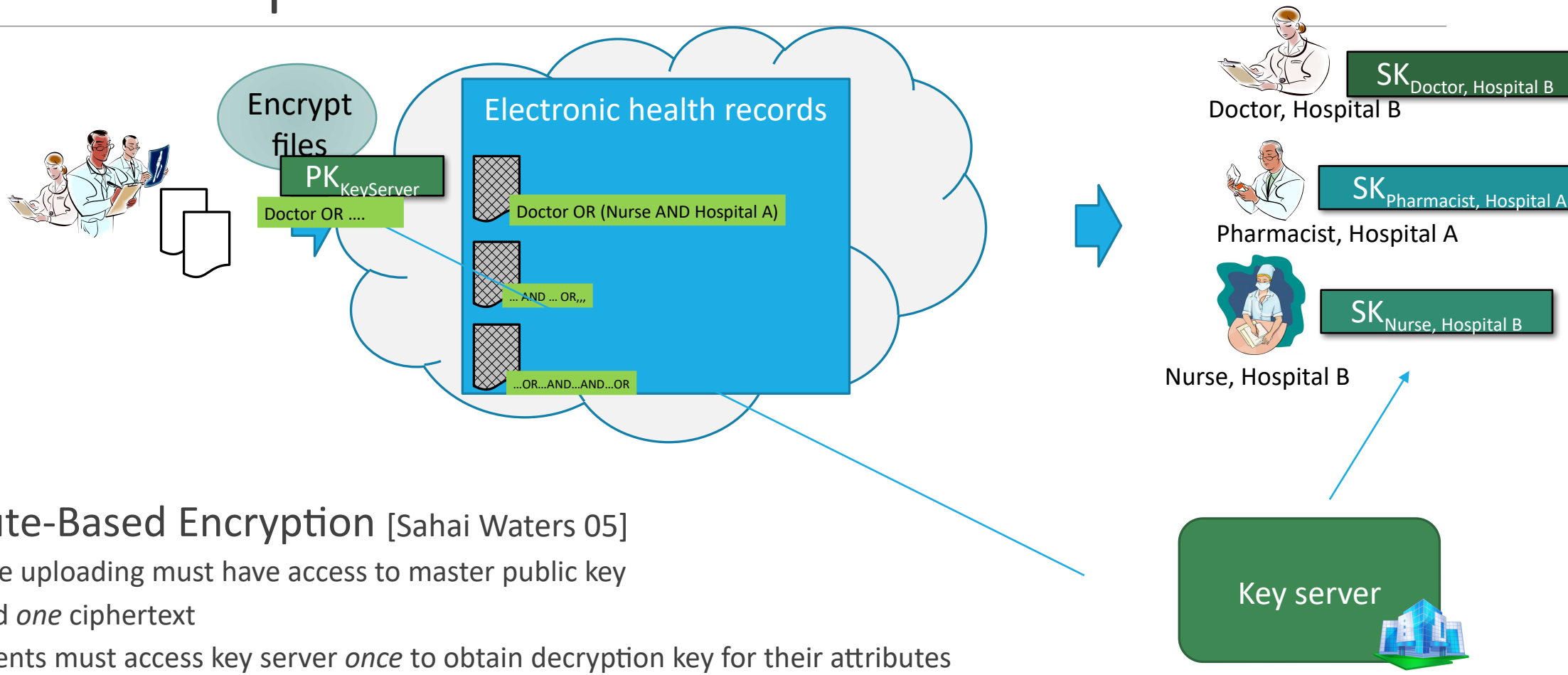


Can we build this from public key encryption?

Solution 4: Encrypt file under public key of trusted key server which checks policy before decrypting

- Recipients must access key server every time they want to decrypt a new file

An example



Attribute-Based Encryption [Sahai Waters 05]

- Anyone uploading must have access to master public key
- Upload *one* ciphertext
- Recipients must access key server *once* to obtain decryption key for their attributes
- Collusion-resistant: 2 recipients together can't decrypt more than they could individually

Overview

What is Attribute Based Encryption (ABE)?

Applications

Variants

Constructing ABE from pairings

Application: Cloudflare's Geo Key Manager v2

Cloudflare stores customer TLS keys

- Must be stored encrypted
- Must be accessible to any machine that users connect to
- Must only be accessible to machines satisfying customer's policy

TLS keys are encrypted and stored in a globally distributed database – quickly accessible everywhere

When new machine is brought up, it contacts a central authority, gets an ABE decryption key corresponding to it's attributes

At TLS connection time, machine decrypts the ciphertext and get the TLS key

Other proposed applications

Encrypted Medical Records application proposed by [Akinyele et al 11]

- Also uses key-policy: Key can be encode a policy with specific restrictions on which ciphertexts it can decrypt
- Automatically extract policies from records / hospital policies
- Allows offline access, if encrypted records are available offline

Private Social Network application proposed by [Baden et al 09]

- Each user generates and publishes a master public key
- Alice can encrypt to Bob's "friends" AND "live in town"

Trusted cloud services application proposed by [Santos et al 12]

- "Policy-sealed data": data is encrypted with policy information
- Monitor checks integrity of nodes and gives out decryption keys
- Nodes only need to contact monitor once per boot

Overview

What is Attribute Based Encryption (ABE)?

Applications

Variants

Constructing ABE from pairings

Ciphertext Policy vs Key-policy

Ciphertext Policy [Bethencourt et al 07]:

- Ciphertexts encrypted with policy
- Secret keys for sets of attributes

Key Policy [Goyal et al 06]

- Ciphertexts encrypted with sets of attributes
- Secret keys for policies

Dual Policy [Attrapadung, Imai 09]

- Combines Key Policy and Ciphertext Policy

Security

CPA:

- Adversary can request secret keys of its choice (gives attribute sets/policies)
- Adversary chooses a ciphertext to try to attack (attribute set/policy and message)
- Cannot decrypt unless allowed by ***one*** of the requested keys

Selective security:

- Adversary must choose ciphertext to attack (attribute set/policy) before seeing public or secret keys
- Captures a targeted attack, but not an opportunistic attacker

CCA:

- Add decryption oracle
- Generic [Yamada et al 11] and optimized constructions

Encoding attributes

Large universe: attributes can be any string

- Early work restricted to fixed attribute set: “small universe”

Attribute sets: {Nurse, Hospital B}

Label-value pairs [Okamoto Takashima 10]

- {Role: Nurse, Hospital: B}

Strings over

Policies are monotonic Boolean formulae [Goyal et al 06]

- (AND and OR gates)
- More generally MSP (e.g. threshold gates)

Allows for non-monotonic formulae

- E.g. Role: NOT Nurse

Complex policies

- DFAs [Waters 12]
- Branching programs [Chen Gay Wee 15]
- (limited) NFAs [Gong Wee 20]

Multiple Authorities

Basic ABE: one trusted authority

- posts public key for encryption
- Issues decryption keys

Multi Authority[Chase07] \ Decentralized ABE [LewkoWaters11](CP-ABE)

- Each authority controls a subset of attributes
  - Posts public keys for those attributes
  - Gives out decryption keys for those attributes
- All of user's secret keys are tied to one *global id,* to prevent collusion
- Encryptor can choose to include any combination of these attributes in their policy
- Security holds even if some authorities are malicious

Other variants

Outsourced Decryption [Green, Hohenberger, Waters 11]

- User can generate a “transformation key” from his secret key
- Cloud can use transformation key to convert ABE ciphertexts to El Gamal ciphertexts
- Reduces cost for user to download and decrypt

ABE with delegation [Goyal et al 06]

- Key policy
- Key can be used to generate a new key for a more restrictive formula

Compact ciphertexts or keys [Attrapadung, Libert, Panafieu 11]

- Normally ciphertext/key length is linear in size of attribute set/policy
- Here, either ciphertext or key can be made constant size (but not both)

Overview

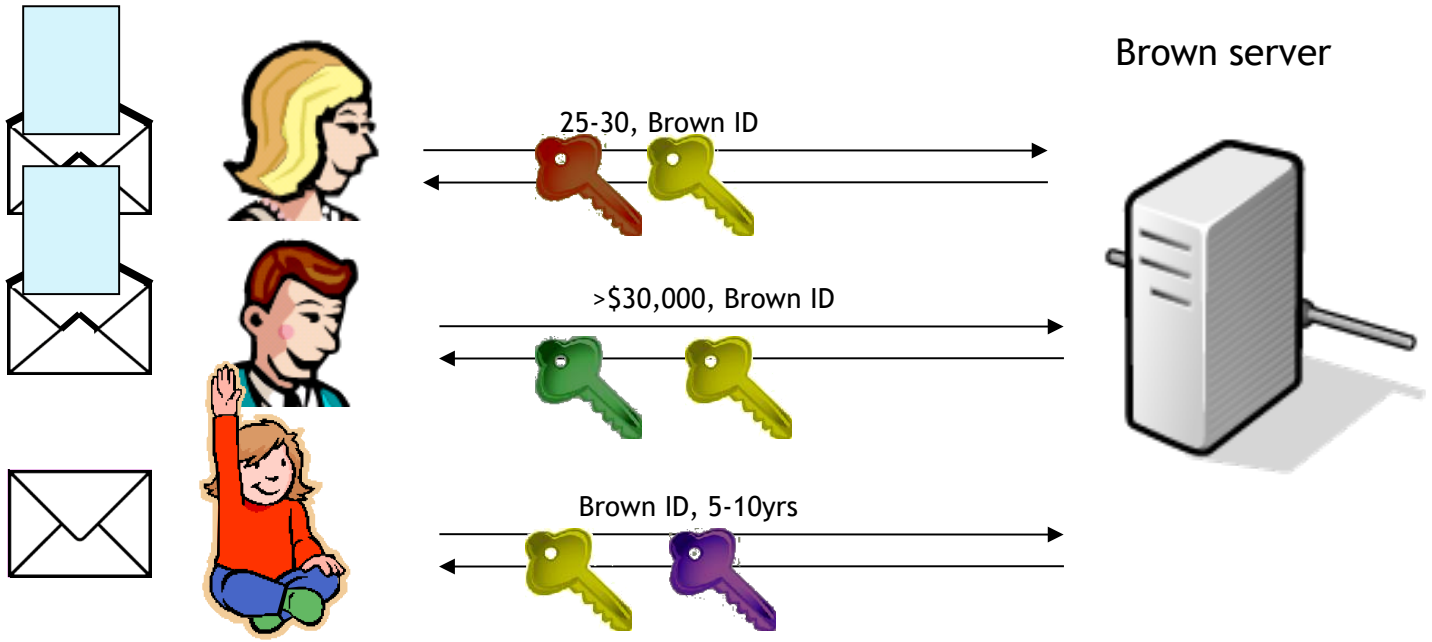
What is Attribute Based Encryption (ABE)?

Applications

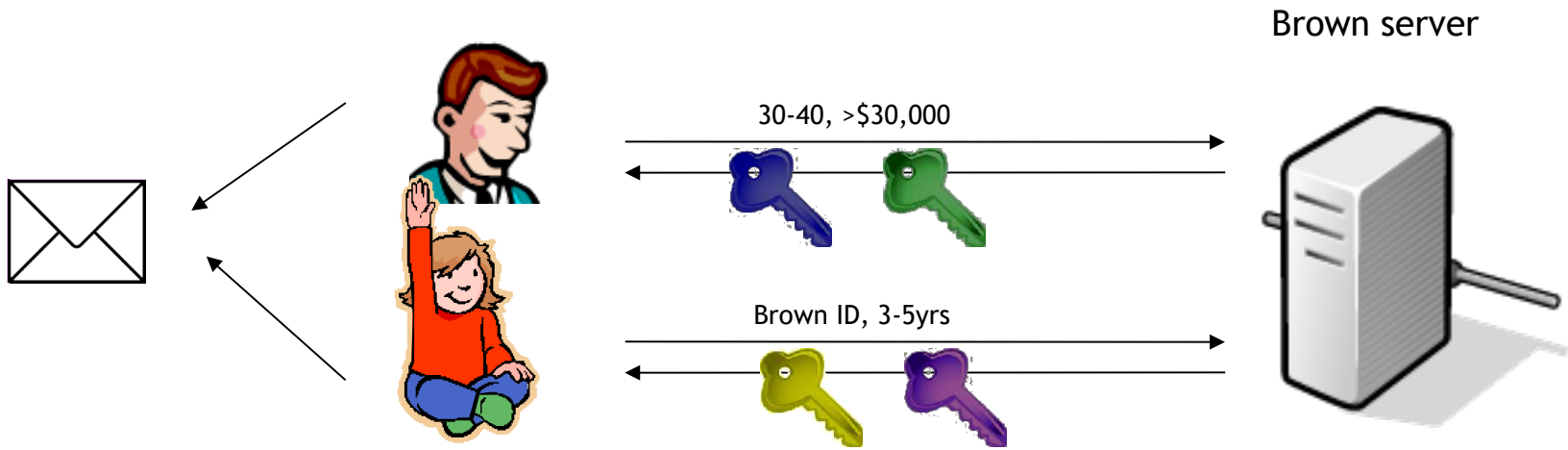
Variants

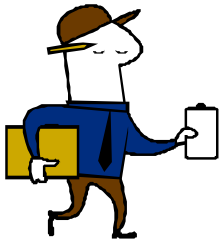
Constructing ABE from pairings

Threshold ABE from pairings [Sahai Waters 05]



Threshold ABE from pairings [Sahai Waters 05]





2 of:

Brown ID

Age 25-30

>$30,000

Tools: Polynomial Secret Sharing

Multiplicative group

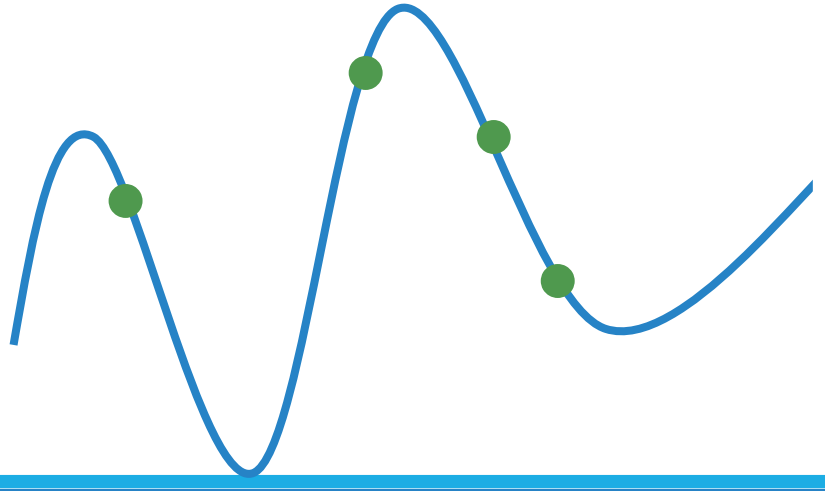
- can
  - multiply
  - divide,
  - raise to integer expt.
- G is element of the group

Given $G^{p(1)}$, $G^{p(2)}$, etc

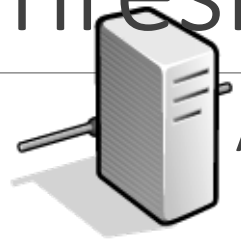
- can reconstruct $G^{p(0)}$

Given $G^{p(1)x}$, $G^{p(2)x}$, etc

- can reconstruct $G^{p(0)x}$



Threshold ABE from pairings [Sahai Waters 05]

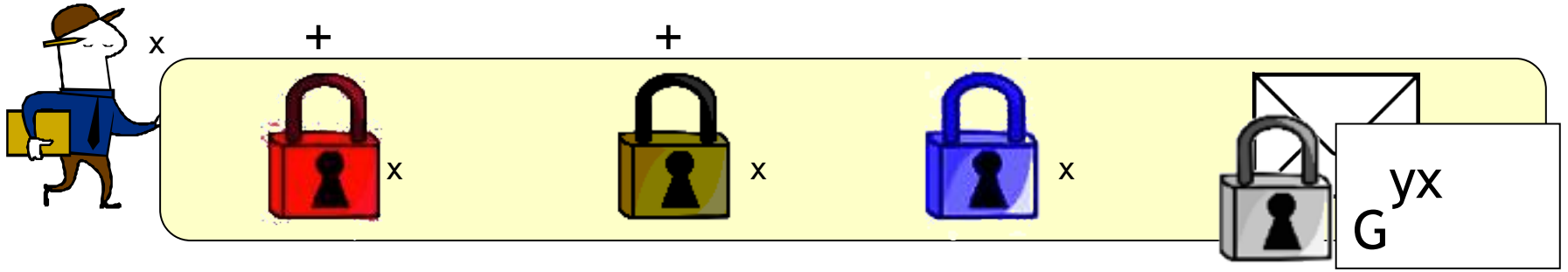


Authority:

SK: y PK: G^{y}

Assume threshold is fixed for the system at 2.

Choose polynomial p for this user, p(0)=y, p has degree 1.



Different p for each user!

$G^{p(1)x}$ $G^{p(2)x}$ ➡ $G^{p(0)x=yx}$

Tools: Bilinear Group

Groups based on elliptic curves

$\boldsymbol{G_1}$, $\boldsymbol{G_2}$, $\boldsymbol{G_T}$

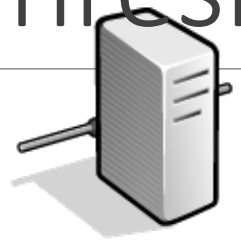
g in $\boldsymbol{G_1}$, h in $\boldsymbol{G_2}$, G in $\boldsymbol{G_T}$

Function e: $\boldsymbol{G_1} \times \boldsymbol{G_2} \rightarrow \boldsymbol{G_T}$

e(g, h)=G

$e(g^a, h^b)=G^{ab}$ for all integers a,b

Threshold ABE from pairings [Sahai Waters 05]



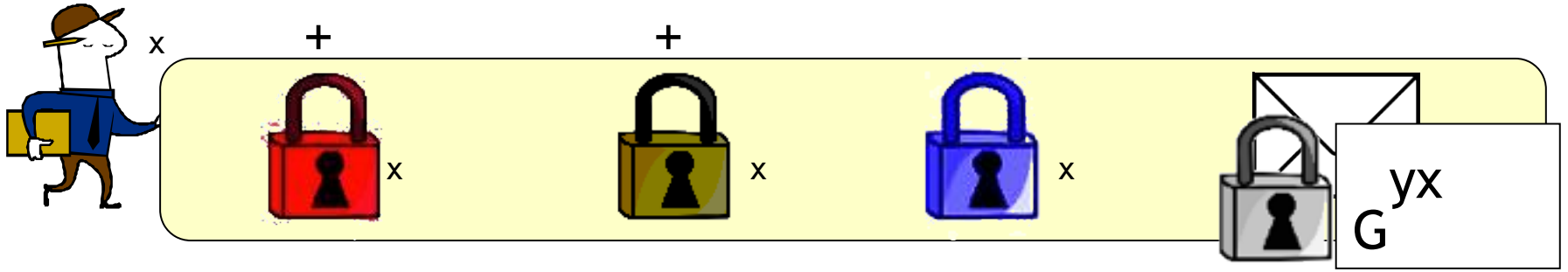
Authority:

SK: y PK: G^{y}

$T1 = g^{t_1}$ $T2 = g^{t_2}$ $T3 = g^{t_3}$

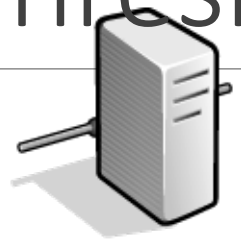
Choose polynomial p for this user,

p(0)=y



$G^{p(1)x}$ $G^{p(2)x}$ ➡ $G^{p(0)x=yx}$

Threshold ABE from pairings [Sahai Waters 05]



Authority:

SK: y PK: G^{y}

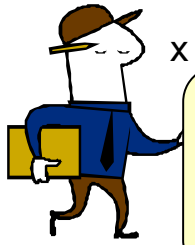
$T1 = g^{t_1}$ $T2 = g^{t_2}$ $T3 = g^{t_3}$

$h^{p(1)/t_1}$

$h^{p(2)/t_2}$

Choose polynomial p for this user,

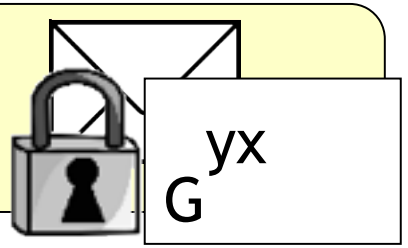
p(0)=y



x

+ +

$T1^{x}$ $T2^{x}$ $T3^{x}$

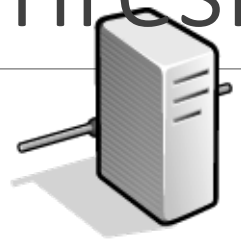


G^{yx}

$G^{p(1)x}$ $G^{p(2)x}$

$G^{p(0)x=yx}$

Threshold ABE from pairings [Sahai Waters 05]



Authority:

SK: y PK: G^{y}

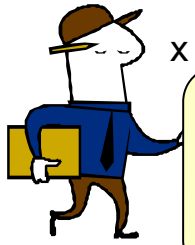
$T1 = g^{t_1}$ $T2 = g^{t_2}$ $T3 = g^{t_3}$

$h^{p(1)/t_1}$

$h^{p(2)/t_2}$

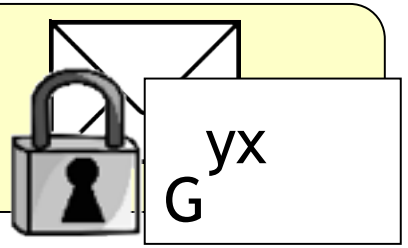
Choose polynomial p for this user,

p(0)=y



x + +

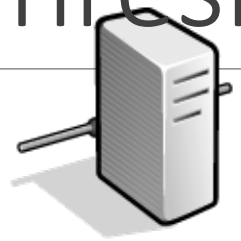
$T1^{x}$ $T2^{x}$ $T3^{x}$



G^{yx}

$e(T1^{x}, h^{p(1)/t_1})$ $G^{p(2)x}$ ➡ $G^{p(0)x=yx}$

Threshold ABE from pairings [Sahai Waters 05]



Authority:

SK: y PK: G^{y}

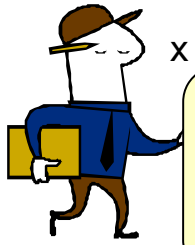
$T1 = g^{t_1}$ $T2 = g^{t_2}$ $T3 = g^{t_3}$

$h^{p(1)/t_1}$

$h^{p(2)/t_2}$

Choose polynomial p for this user,

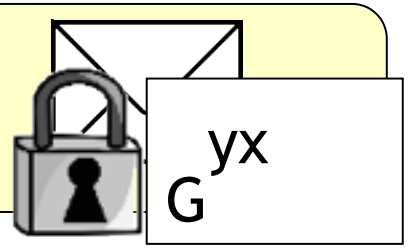
p(0)=y



x

+ +

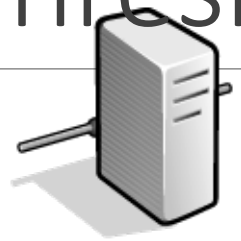
$T1^{x}$ $T2^{x}$ $T3^{x}$



G^{yx}

$e(g^{t_1 x}, h^{p(1)/t_1})$ $G^{p(2)x}$ ➡ $G^{p(0)x} = G^{yx}$

Threshold ABE from pairings [Sahai Waters 05]



Authority:

SK: y PK: G^{y}

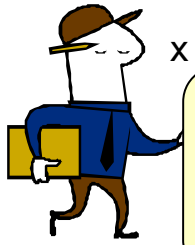
$T1 = g^{t_1}$ $T2 = g^{t_2}$ $T3 = g^{t_3}$

$h^{p(1)/t_1}$

$h^{p(2)/t_2}$

Choose polynomial p for this user,

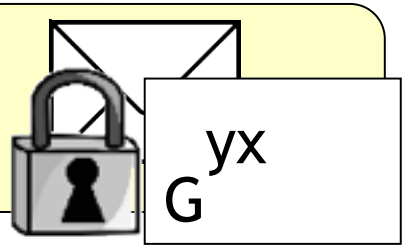
p(0)=y



x

+ +

$T1^{x}$ $T2^{x}$ $T3^{x}$

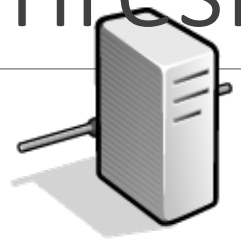


G^{yx}

$G^{t_1 x \cdot p(1)/t_1}$ $G^{p(2)x}$

$G^{p(0)x=yx}$

Threshold ABE from pairings [Sahai Waters 05]



Authority:

SK: y PK: G^{y}

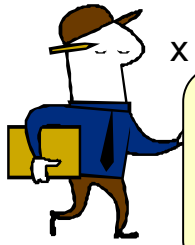
$T1 = g^{t_1}$ $T2 = g^{t_2}$ $T3 = g^{t_3}$

$h^{p(1)/t_1}$

$h^{p(2)/t_2}$

Choose polynomial p for this user,

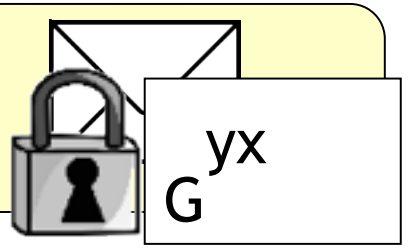
p(0)=y



x

+ +

$T1^{x}$ $T2^{x}$ $T3^{x}$



G^{yx}

$G^{p(1)x}$ $G^{p(2)x}$

$G^{p(0)x=yx}$

Constructing ABE

Most efficient schemes: tailored constructions secure in the GGM model

Alternatively, a general design approach

1) Design a scheme that works for 1 ciphertext and 1 key
   - Correct: If policy is satisfied, decryption works
   - Not trivially broken: If policy is not satisfied, no linear combination of key*ciphertext components gives the message
2) Apply a transformation (Pair encodings [Wee14], Predicate encodings [Attrapadung 14], Symbolic security [Agrawal Chase 17]):
   - Increases ciphertext/secret key size a bit
   - Adds collusion resistance and full security against any attack
   - Security under SXDH/DLIN or q-type assumptions

ABE: current status

Fairly mature constructions

- Efficient, flexible, realistic security models

Just beginning to be deployed

- Other compelling applications?
- Other desirable properties?

Post-quantum security

- Some applications this may be less important
- Lattice-based constructions

Questions