ANTI-Fraud: ABE Solution

CifrisCloud – Cyptography for the Cloud

September 2024 NTT DATA Italia at CIFRIS24

Overview

1. Cryptography & NTT Group

2. The Banking Sector

3. Attribute-Based Encryption
   - Key-Policy Attribute-Based Encryption

4. ANTI-fraud ABE
   - The Demonstration

Cryptography & NTT Group

September 2024 NTT DATA Italia at CIFRIS24

Cryptography & NTT Group

Brent Waters is the **Director of the Cryptography & Information Security Laboratories (CIS Lab)**, a Distinguished Scientist at **NTT Research** and a professor at the University of Texas.

Elette Boyle is a **Senior Scientist** in the **CIS Lab** at **NTT Research** and an Associate Professor in the Efi Arazi School.

Martina Palmucci is a Cybersecurity System Engineer at **NTT DATA Italia**.

The Banking Sector

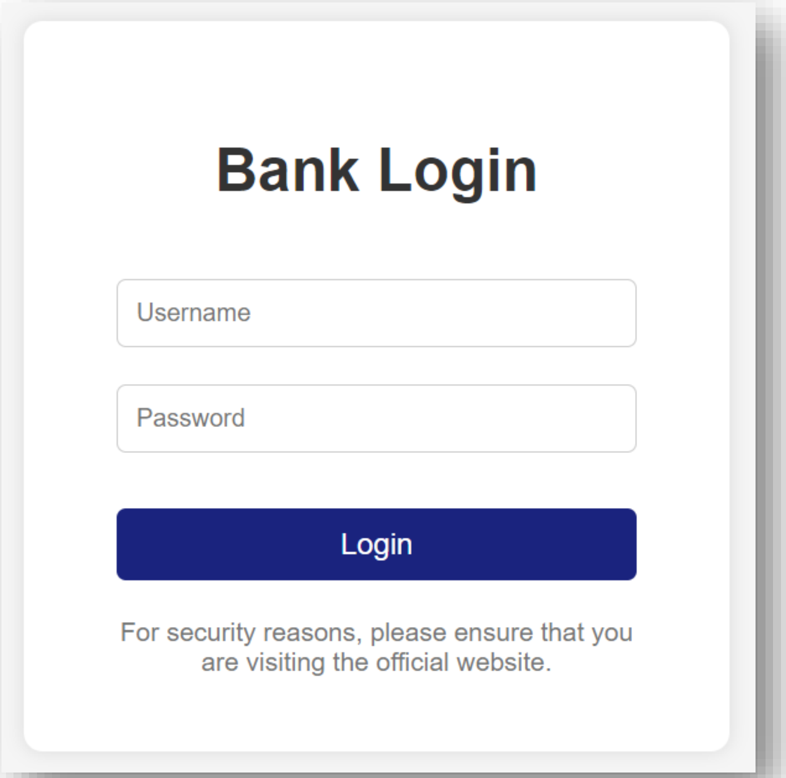
The scenario

Risk-Based Authentication

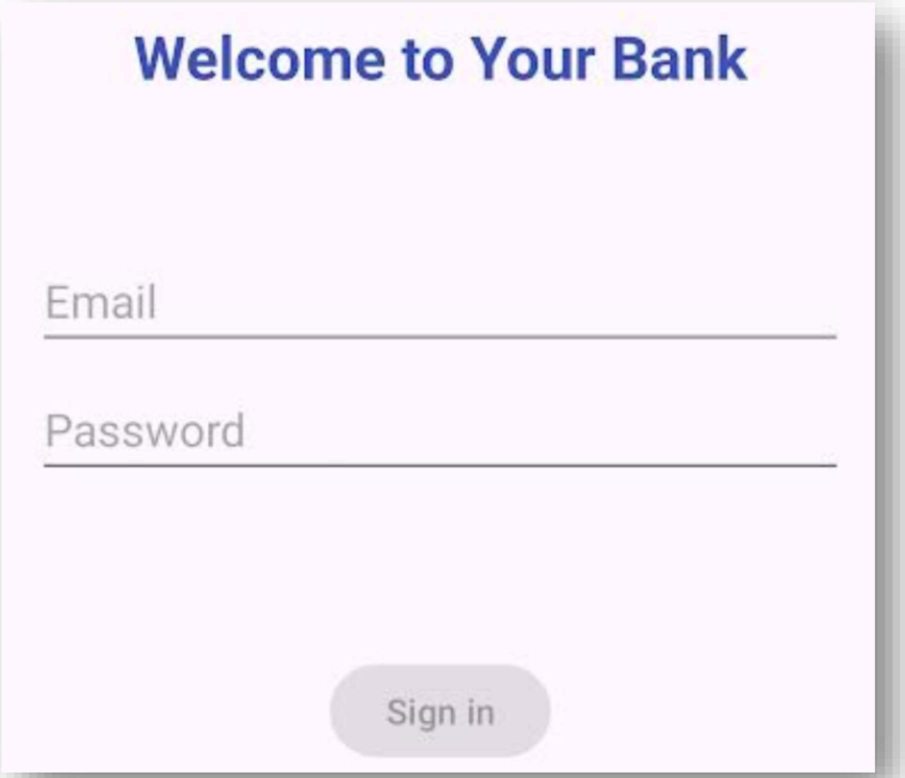
- **Banking transactions** authorised based on risk level
- **Risk level** assessed according to **standard risk policies**
- **PSD2 regulation:** Dynamic Linking between the amount and the recipient of the payment
- **User privacy preserving**

Spoiler Alert!!

Web App



Mobile App



Attribute-Based Encryption

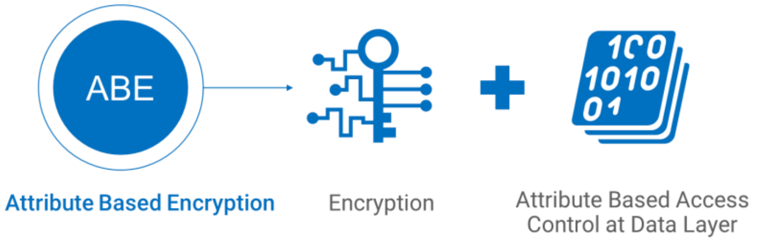
The cryptographic paradigm

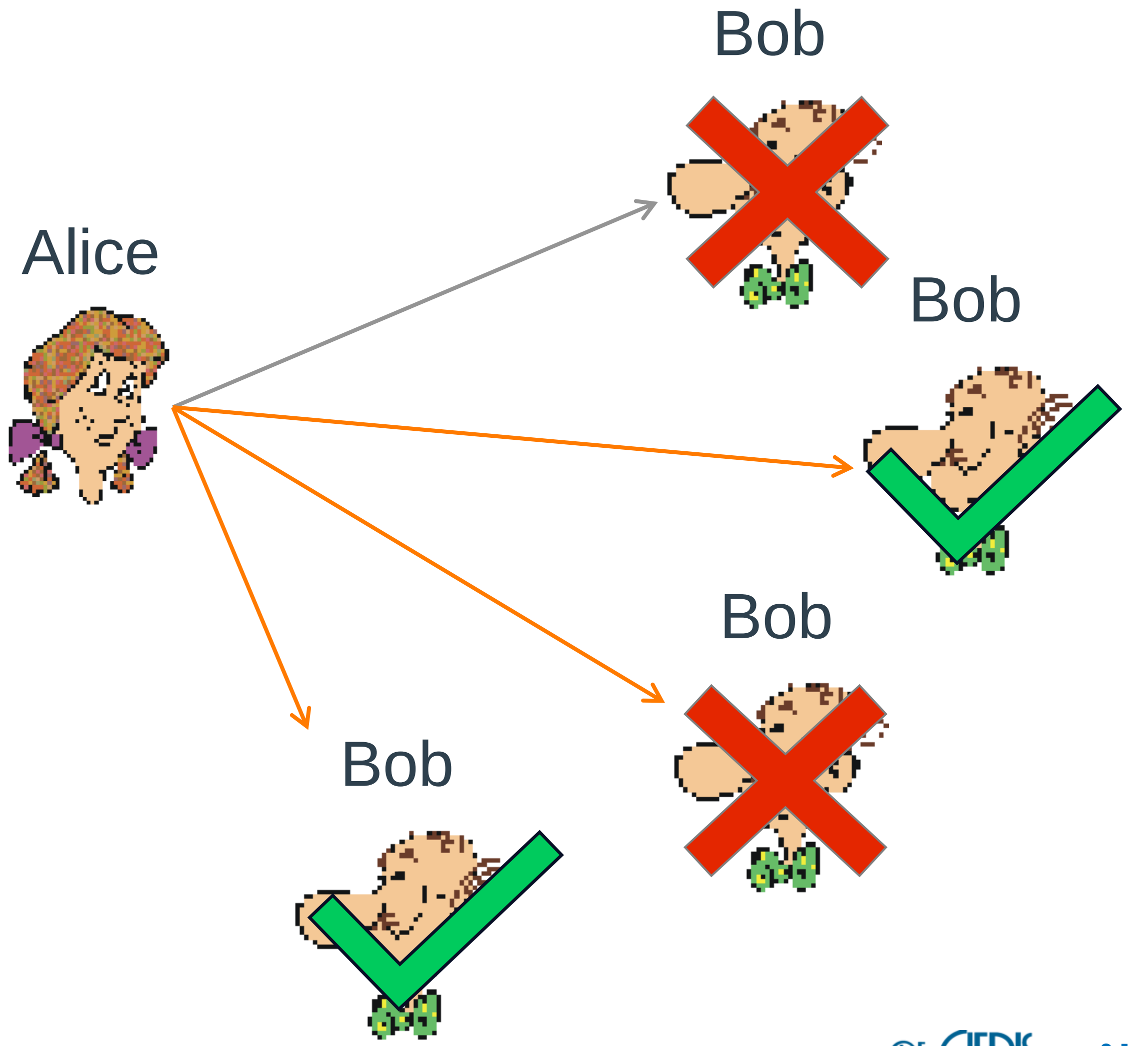
September 2024 NTT DATA Italia at CIFRIS24

Attribute-Based Encryption (ABE)

Attribute-Based Encryption (ABE) is an innovative asymmetric cryptography paradigm that introduces one to many encryption.

It enables very granular definition of data access permissions by embedding their control directly into cryptographic functions.





Attribute-Based Encryption (ABE)

Attribute-Based Encryption (ABE) is an innovative asymmetric cryptography paradigm that introduces one to many encryption.

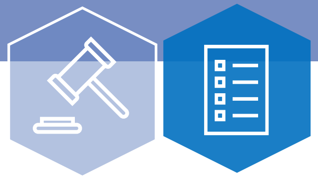
It enables very granular definition of data access permissions by embedding their control directly into cryptographic functions.

Key

Ciphertext

The mechanism is based on the following elements:

- A list of attributes (numeric, Boolean or string values)
- A policy (Boolean expression built on top of the attributes)



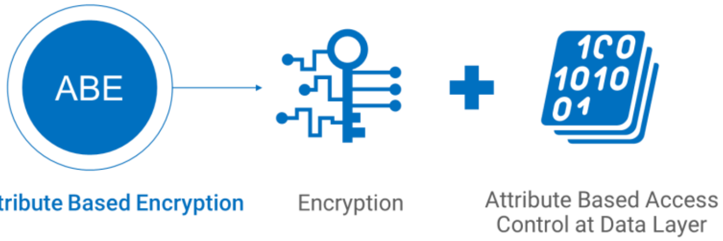
Policy

Set of attributes

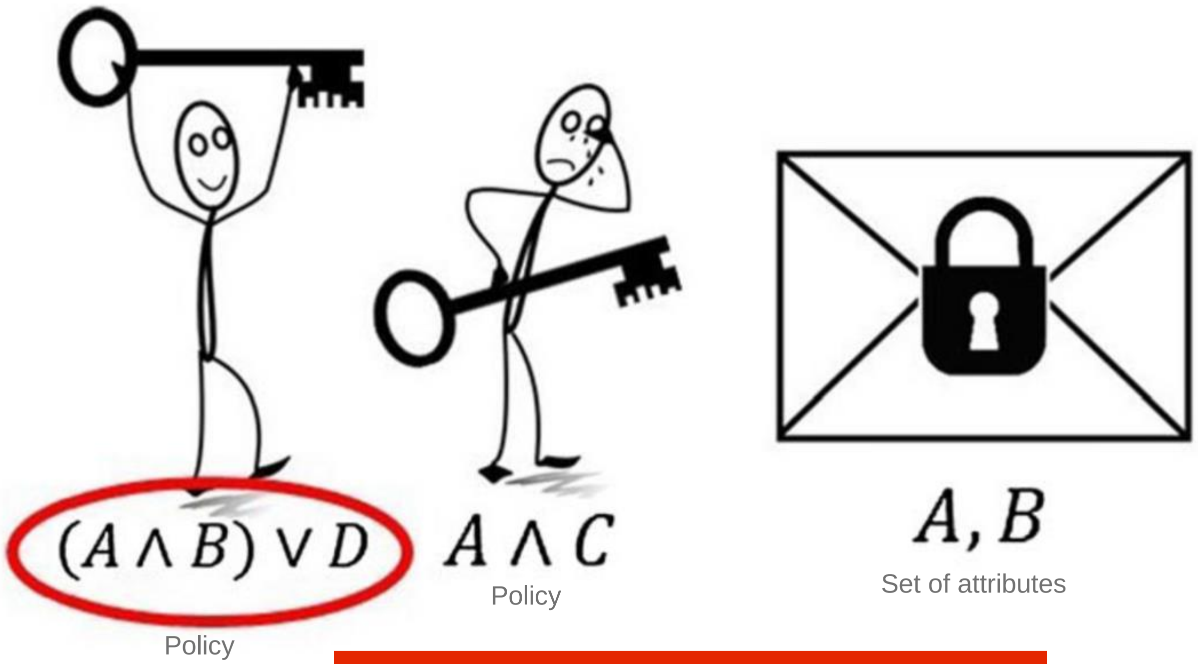
In such a system,
data can be decrypted only if

the list of attributes forged in the key [or ciphertext]

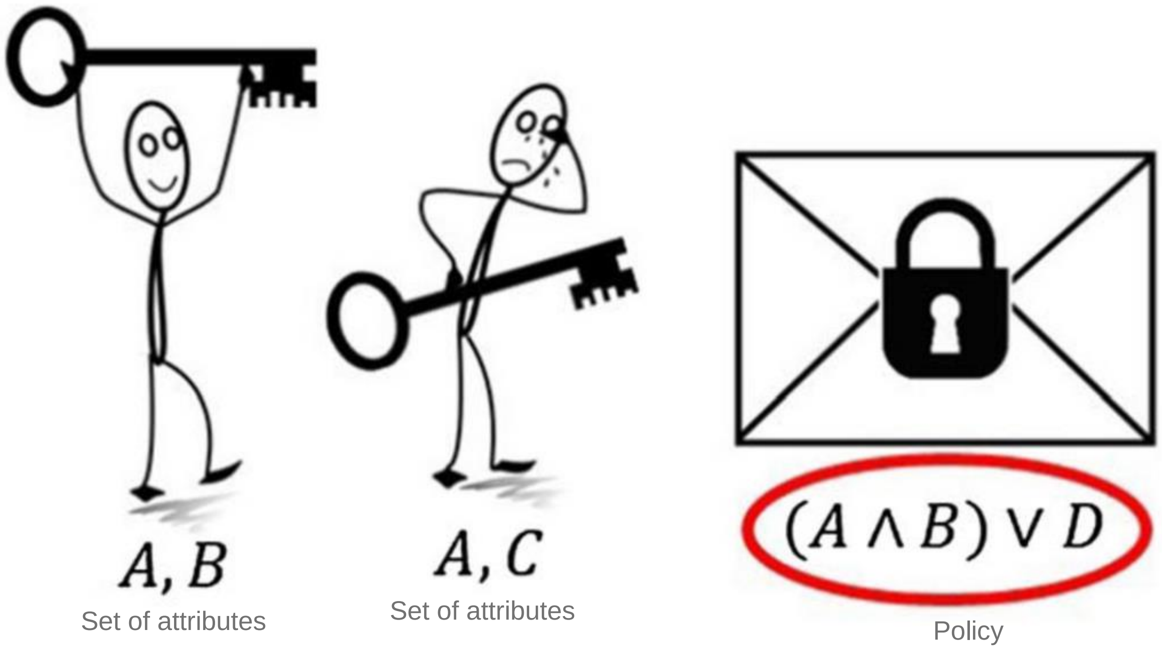
satisfies the policy forged in the ciphertext [or key].



Policy, Attributes & 2 Types of ABE



Key-Policy
Attribute-Based Encryption
(KP-ABE)



Ciphertext-Policy
Attribute-Based Encryption
(CP-ABE)

Key-Policy ABE by Goyal et al.

The ABE scheme

KP-ABE scheme

Authority Setup
Setup(λ)
= MK, PK

Key Generation
KeyGen(MK, P)
= SK

Encryption
E(PK, A, m) = c

Decryption
D(SK, c) = m

Legend

λ: security parameter

MK: master private key
PK: master public key

P: access policy
SK: user secret key

A: (list of) attributes
m: plaintext
C: ciphertext

KP-ABE Goyal et al. scheme – Background

- **Monotone Access Structure**

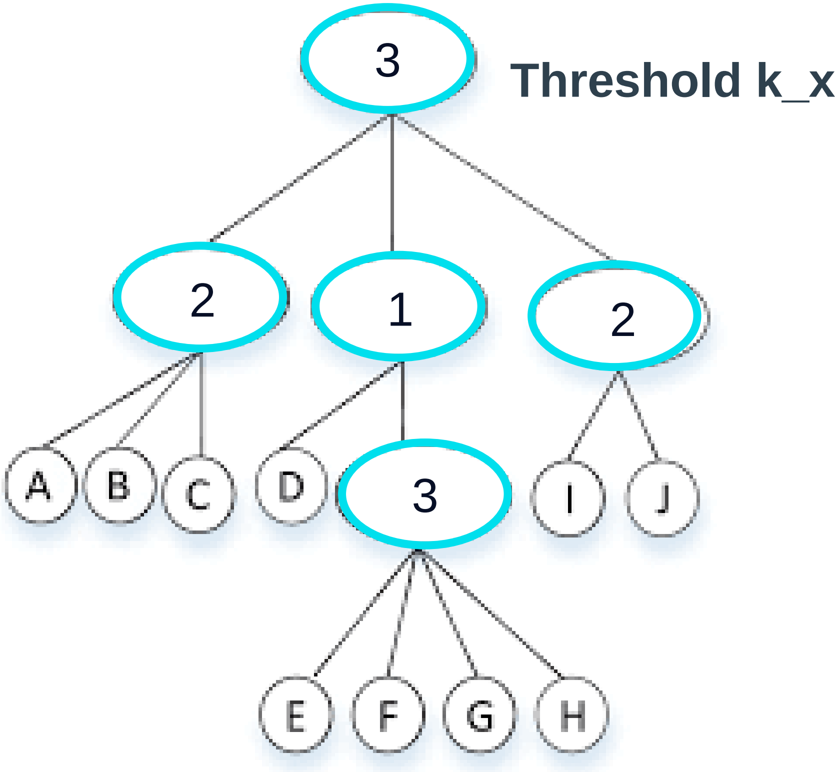
Definition 1 (Access Structure [4]) *Let* $\{P_1, P_2, \ldots, P_n\}$ *be a set of parties. A collection* $\mathbb{A} \subseteq 2^{\{P_1,P_2,\ldots,P_n\}}$ *is monotone if* $\forall B, C$: if $B \in \mathbb{A}$ *and* $B \subseteq C$ *then* $C \in \mathbb{A}$. *An* access structure *(respectively, monotone access structure) is a collection (respectively, monotone collection)* $\mathbb{A}$ *of non-empty subsets of* $\{P_1, P_2, \ldots, P_n\}$, *i.e.,* $\mathbb{A} \subseteq 2^{\{P_1,P_2,\ldots,P_n\}} \setminus \{\emptyset\}$. *The sets in* $\mathbb{A}$ *are called the* authorized sets, *and the sets not in* $\mathbb{A}$ *are called the* unauthorized sets.

- The role of the **parties** is taken by the **attributes**.
- The **access structure A** will contain the **authorized sets of attributes**.
- The **number of attributes** in the system **will not be doubled**.
- The **access structure A** will be represented as an **access tree** $\mathscr{T}$ and realized with a ***Linear Secret-Sharing Scheme* (LSSS)**.

KP-ABE Goyal et al. scheme – Background

- **Access Tree**

- The **access structure A** will be represented as an **access tree $\mathscr{T}$** and realized with a ***Linear Secret-Sharing Scheme* (LSSS)**.



KP-ABE Goyal et al. scheme – Background

- **Bilinear map**

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two multiplicative cyclic groups of prime order p. Let g be a generator of $\mathbb{G}_1$ and e be a bilinear map, $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. The bilinear map e has the following properties:

1. Bilinearity: for all $u, v \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
2. Non-degeneracy: $e(g, g) \neq 1$.

We say that $\mathbb{G}_1$ is a bilinear group if the group operation in $\mathbb{G}_1$ and the bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ are both efficiently computable. Notice that the map e is symmetric since $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.

KP-ABE Goyal et al. scheme – The construction

- Setup

Setup Define the universe of attributes $\mathcal{U} = \{1, 2, \ldots, n\}$. Now, for each attribute $i \in \mathcal{U}$, choose a number t_i uniformly at random from $\mathbb{Z}_p$. Finally, choose y uniformly at random in $\mathbb{Z}_p$. The published public parameters PK are

$$T_1 = g^{t_1}, \ldots, T_{|\mathcal{U}|} = g^{t_{|\mathcal{U}|}}, Y = e(g, g)^y .$$

The master key MK is:

$$t_1, \ldots, t_{|\mathcal{U}|}, y .$$

KP-ABE Goyal et al. scheme – The construction

- **Encryption**

Encryption (M, γ, PK) To encrypt a message $M \in \mathbb{G}_2$ under a set of attributes γ, choose a random value $s \in \mathbb{Z}_p$ and publish the ciphertext as:

$$E = (\gamma, E' = MY^s, \{E_i = T_i^s\}_{i \in \gamma}) \ .$$

KP-ABE Goyal et al. scheme – The construction

- **Key Generation**

Polynomials

Degree

Constant term & Coeff.

Key Generation $(\mathcal{T}, \mathrm{MK})$ The algorithm outputs a key that enables the user to decrypt a message encrypted under a set of attributes γ if and only if $\mathcal{T}(\gamma) = 1$. The algorithm proceeds as follows. First choose a polynomial q_x for each node x (including the leaves) in the tree $\mathcal{T}$. These polynomials are chosen in the following way in a top-down manner, starting from the root node r.

For each node x in the tree, set the degree d_x of the polynomial q_x to be one less than the threshold value k_x of that node, that is, $d_x = k_x - 1$. Now, for the root node r, set $q_r(0) = y$ and d_r other points of the polynomial q_r randomly to define it completely. For any other node x, set $q_x(0) = q_{\mathrm{parent}(x)}(\mathrm{index}(x))$ and choose d_x other points randomly to completely define q_x.

Once the polynomials have been decided, for each leaf node x, we give the following secret value to the user:

$$D_x = g^{\frac{q_x(0)}{t_i}} \quad \text{where } i = \mathrm{att}(x) .$$

The set of above secret values is the decryption key D.

KP-ABE Goyal et al. scheme – The construction

- **Decryption**

Recursive algorithm from leaves to the root

Decryption (E, D) We specify our decryption procedure as a recursive algorithm . For ease of exposition we present the simplest form of the decryption algorithm and discuss potential performance improvements in the next subsection.

We first define a recursive algorithm $\mathrm{DecryptNode}(E, D, x)$ that takes as input the ciphertext $E = (\gamma, E', \{E_i\}_{i\in\gamma})$, the private key D (we assume the access tree $\mathcal{T}$ is embedded in the private key), and a node x in the tree. It outputs a group element of $\mathbb{G}_2$ or $\perp$.

If leaf node:

Let $i = \mathrm{att}(x)$. If the node x is a leaf node then:

$$\mathrm{DecryptNode}(E, D, x) = \begin{cases} e(D_x, E_i) = e(g^{\frac{q_x(0)}{t_i}}, g^{s\cdot t_i}) = e(g,g)^{s\cdot q_x(0)} & \text{if } i \in \gamma \\ \perp \text{ otherwise} \end{cases}$$

If branch node:

1. save previous results

We now consider the recursive case when x is a non-leaf node. The algorithm $\mathrm{DecryptNode}(E, D, x)$ then proceeds as follows: For all nodes z that are children of x, it calls $\mathrm{DecryptNode}(E, D, z)$ and stores the output as F_z. Let S_x be an arbitrary k_x-sized set of child nodes z such that $F_z \neq \perp$. If no such set exists then the node was not satisfied and the function returns $\perp$.

KP-ABE Goyal et al. scheme – The construction

- **Decryption**

We also define the Lagrange coefficient $\Delta_{i,S}$ for $i \in \mathbb{Z}_p$ and a set, S, of elements in $\mathbb{Z}_p$: $\Delta_{i,S}(x) = \prod_{j \in S, j \neq i} \frac{x-j}{i-j}$. We will associate each attribute with a unique element in $\mathbb{Z}_p^*$.

Otherwise, we compute:

2. compute

$$
\begin{aligned}
F_x &= \prod_{z \in S_x} F_z^{\Delta_{i,S'_x}(0)}, \qquad \text{where } {}^{i=\text{index}(z)}_{S'_x=\{\text{index}(z):z\in S_x\}} \\
&= \prod_{z \in S_x} \left(e(g,g)^{s \cdot q_z(0)}\right)^{\Delta_{i,S'_x}(0)} \\
&= \prod_{z \in S_x} \left(e(g,g)^{s \cdot q_{\text{parent}(z)}(\text{index}(z))}\right)^{\Delta_{i,S'_x}(0)} \qquad \text{(by construction)} \\
&= \prod_{z \in S_x} e(g,g)^{s \cdot q_x(i) \cdot \Delta_{i,S'_x}(0)} \\
&= e(g,g)^{s \cdot q_x(0)} \qquad \text{(using polynomial interpolation)}
\end{aligned}
$$

and return the result.

KP-ABE Goyal et al. scheme – The construction

- **Decryption**

If root:

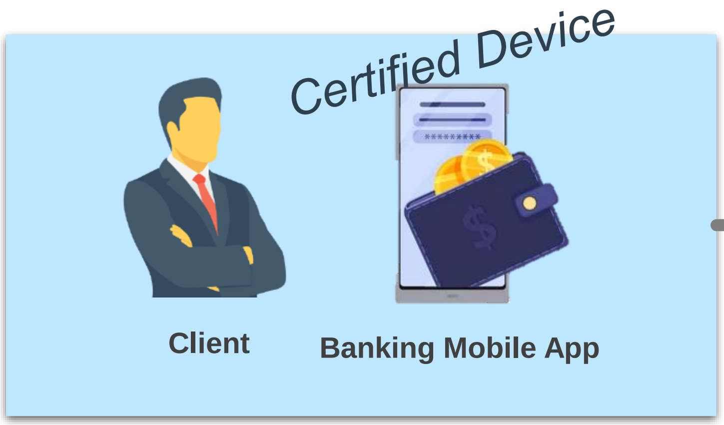
Now that we have defined our function DecryptNode, the decryption algorithm simply calls the function on the root of the tree. We observe that $\mathrm{DecryptNode}(E, D, r) = e(g, g)^{ys} = Y^s$ if and only if the ciphertext satisfies the tree. Since, $E' = MY^s$ the decryption algorithm simply divides out Y^s and recovers the message M.

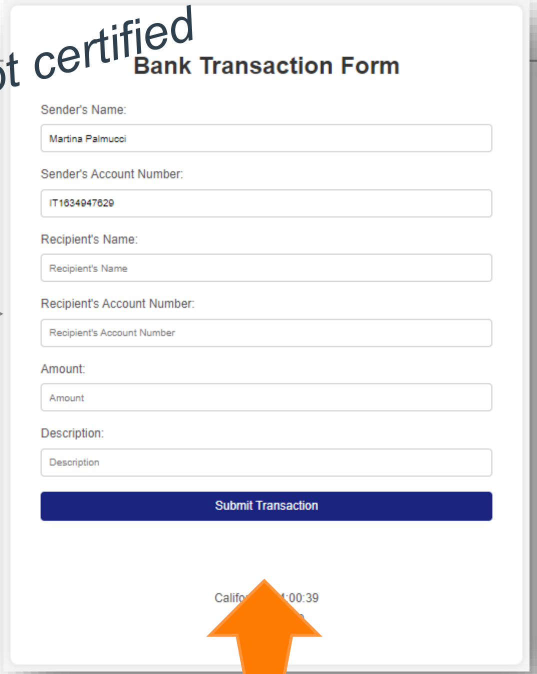
We recovered the message!

ANTI-fraud ABE

The application

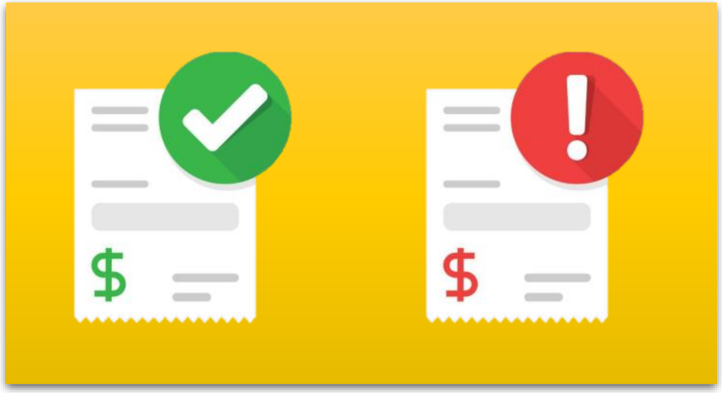
KP-ABE applied to a Bank Transaction





Bank Transaction Form

Sender's Name:
Martina Palmucci
Sender's Account Number:
IT1634947629
Recipient's Name:
Recipient's Name
Recipient's Account Number:
Recipient's Account Number
Amount:
Amount
Description:
Description
Submit Transaction



Key

with

Policy
= Bank Client Behavior

Ciphertext

with

List of attributes
= Bank Transfer features

Does **List of Attributes match Policy**?

Transaction Success

Transaction Warning or **Fail**

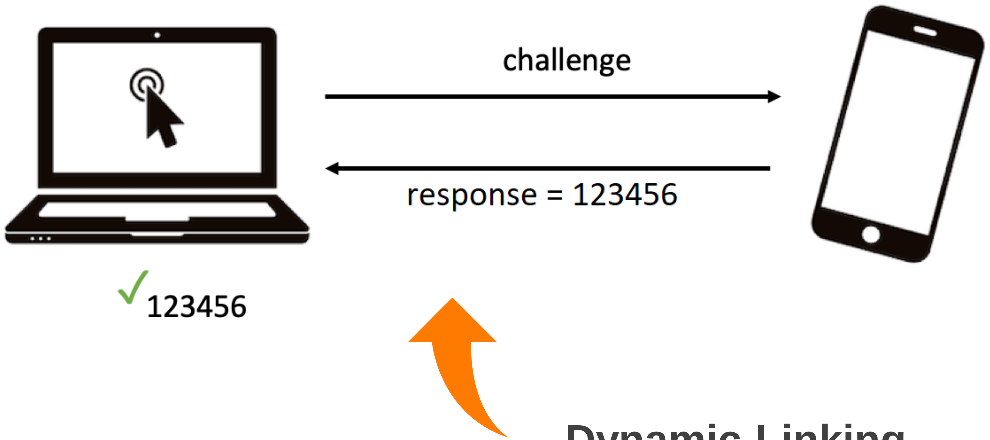
NTT DATA

ABE + Challenge Response

in the banking sector

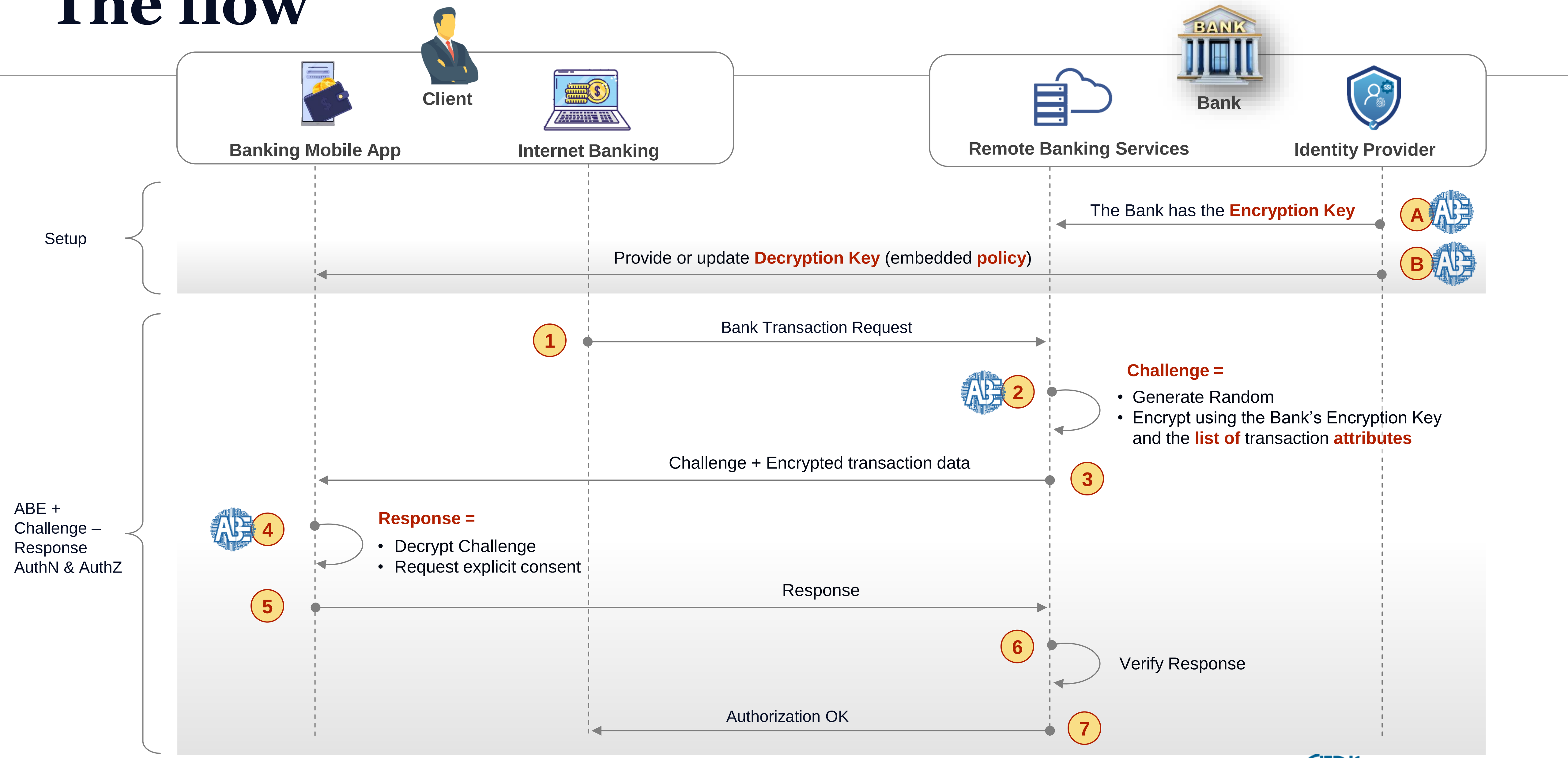
+

Challenge – Response



Dynamic Linking

The flow



Client
Banking Mobile App
Internet Banking
Bank
Remote Banking Services
Identity Provider
Setup
The Bank has the Encryption Key
A
Provide or update Decryption Key (embedded policy)
B
ABE + Challenge – Response AuthN & AuthZ
1
Bank Transaction Request
2
Challenge =
• Generate Random
• Encrypt using the Bank's Encryption Key and the list of transaction attributes
Challenge + Encrypted transaction data
3
4
Response =
• Decrypt Challenge
• Request explicit consent
5
Response
6
Verify Response
Authorization OK
7

The Demonstration

La demo: User pools

SENDER POOL

- **Martina Palmucci:** ('martina.palmucci', 'Martina Palmucci', 'IT1634947629', 'Italy')
- **Stephen Curry:** ('stephen.curry', 'Stephen Curry', 'US7583967349', 'USA')

RECEIVER POOL

- **Martina Palmucci**
- **Stephen Curry**
- **Hidetoshi Nakata:** ('hidetoshi.nakata', 'Hidetoshi Nakata', 'JP1162095736', 'Japan'),
- **Kim Jong-un:** ('kim.jongun', 'Kim Jong-un', 'NK6389675810', 'North-Korea'),
- **Vladimir Putin:** ('vladimir.putin', 'Vladimir Putin', 'RS4688121209', 'Russia');

La demo: Successful case

- Sender

STEPHEN CURRY

  – **Amount** <= 100.000 USD
  – Recipient Country
    - Italy
    - USA
    - **Japan**
    - Russia

- Receiver

HIDETOSHI NAKATA

- Home Country: Japan

La demo: Warning case

- Sender

MARTINA PALMUCCI

- **Time:** Working hours (**9am-6pm**)
- **Amount** <= 1.000 USD
- Recipient Country
  - Italy
  - **USA**
  - Japan

- Receiver

STEPHEN CURRY

- Home Country: USA

Grazie a tutti!

NTT DATA