Foundations of Cybersecurity

WORKSHOP ON ATTRIBUTE-BASED ENCRYPTION

Michele La Manna, Ph.D.

OUTLINE

- What is Attribute-Based Encryption (ABE)?
  - CP-ABE & KP-ABE
  - Primitives
  - General Architecture
- Realistic deploy of ABE
  - Example Use-Cases
  - Performance on ESP32
- ABE Mathematical Foundations:
  - Elliptic Curve Cryptography
  - Pairing-based Cryptography
  - In-depth: The Original CP-ABE
- The CP-ABE toolkit
  - How does it work?
  - Exercises

WHAT IS ATTRIBUTE-BASED ENCRYPTION?

- A “novel” Asymmetric Encryption Family of Schemes that features a single-encryption multiple-receiver fine-grained access control mechanism.
- First proposed in “*Fuzzy identity-based encryption*”, Sahai, A. and Waters, B. (2005).

Key concepts:

- **Attribute**: an abstract property that can be associated to a piece of information or to a data consumer.
- **Attribute Set:** the list of attributes that describes either a piece of information or a data consumer.
- **Access Policy:** a boolean formula over some attributes. An Access Policy (more simply, policy) is satisfied by an attribute set if the attributes embedded in such a set evaluated with the policy return TRUE.

TWO TYPES OF ABE

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

Who can access this piece of information?

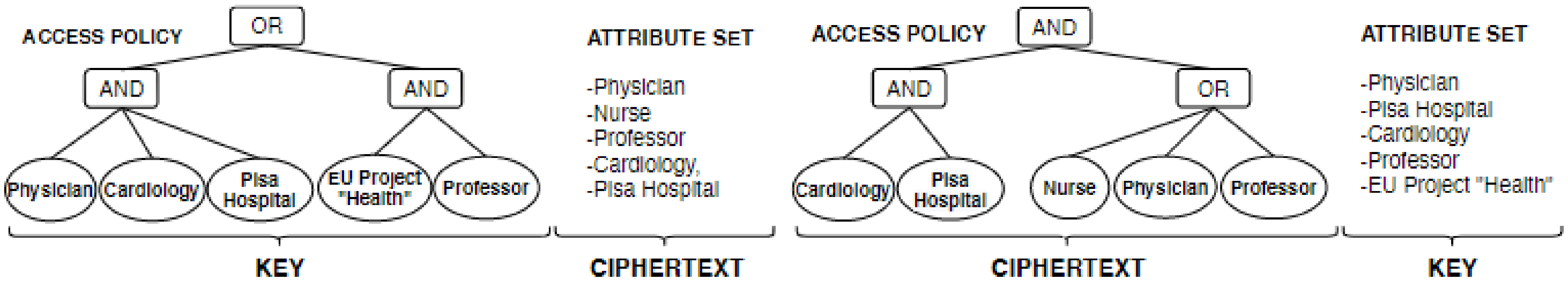
A policy describes the ciphertext containing the piece of information, while an attribute set describes a decryption key.

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

What kind of information can I access?

A policy describes the access rights of a decryption key, while an attribute set describes the piece of information contained in the ciphertext.

KP-ABE AND CP-ABE EXAMPLES



(a) KP-ABE

(b) CP-ABE

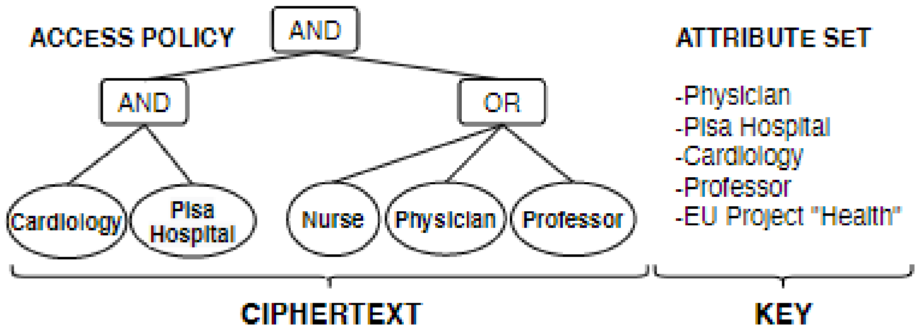
BASIC CP-ABE PRIMITIVES

Setup: Generation of the Public Key and Master Key;

Key Generation: Given in input the Master Key and an Attribute Set, it outputs a Decryption Key;

Encryption: Given in input the Public Key, an Access Policy, and a message, it outputs a ciphertext.

Decryption: Given in input a decryption key and a ciphertext, it outputs the message iff the attribute set embedded in the decryption key satisfies the access policy embedded in the ciphertext.



(b) CP-ABE

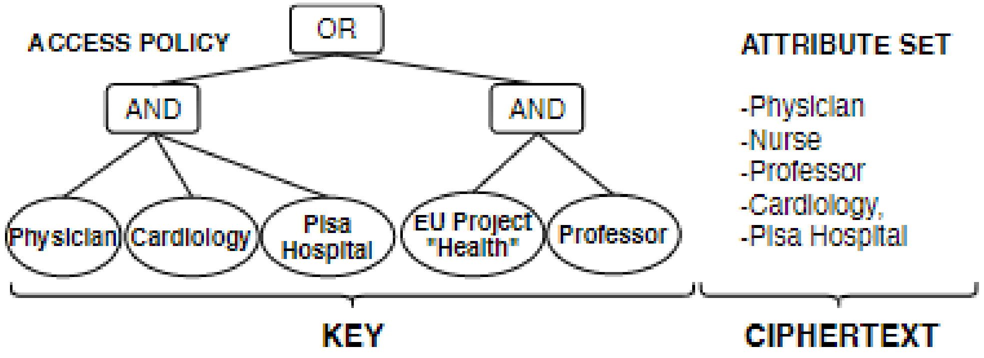
BASIC KP-ABE PRIMITIVES

Setup: Generation of the Public Key and Master Key;

Key Generation: Given in input the Master Key and an Access Policy, it outputs a Decryption Key;

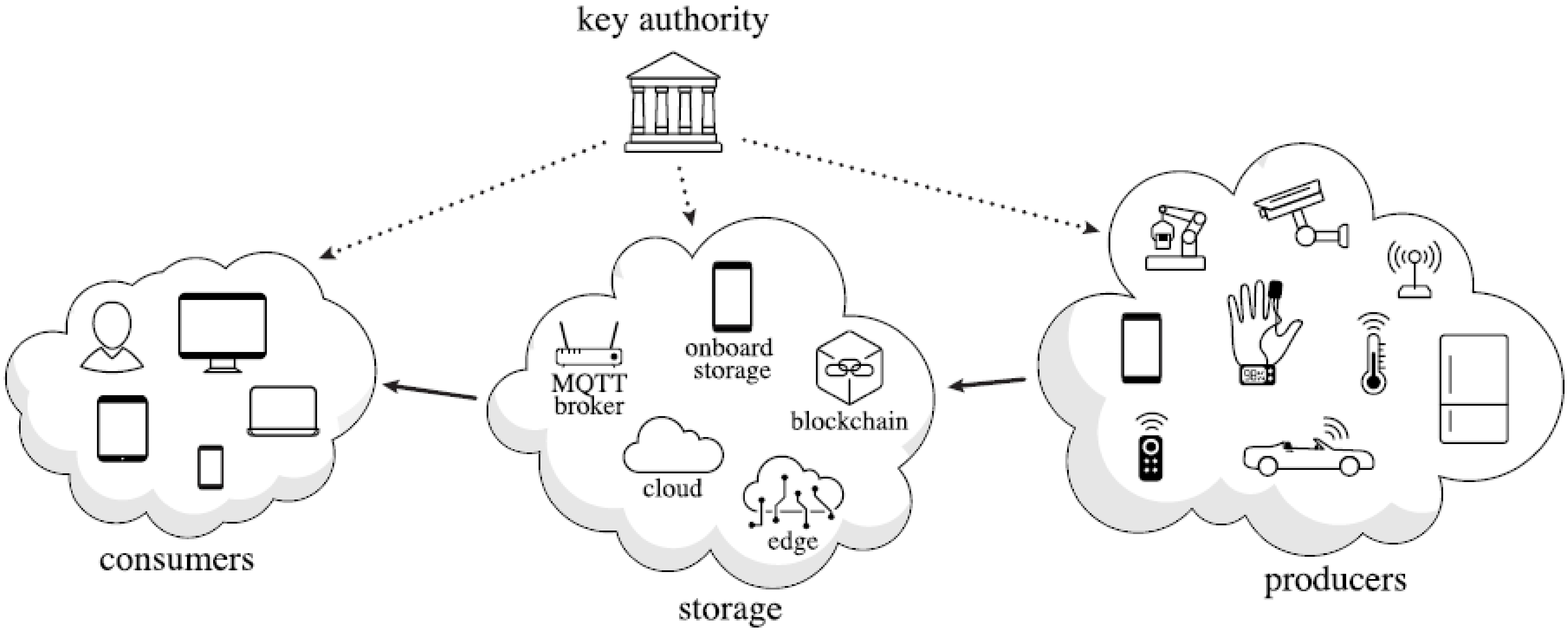
Encryption: Given in input the Public Key, an attribute set, and a message, it outputs a ciphertext.

Decryption: Given in input a decryption key and a ciphertext, it outputs the message iff the attribute set embedded in the ciphertext key satisfies the access policy embedded in the decryption key.



(a) KP-ABE

ABE GENERAL ARCHITECTURE



WHY USE ABE?

Security is one of the most critical issues of Internet communications.
Major cyberthreats detected by OWASP top 10 in 2021:

Sensitive data Exposure:

- Data Breach/Leackage of Cloud Servers
- Related problem: **Data at rest**

Broken Authentication:

- Esfiltration of Private Keys / Passwords
- Related Problem: **Key Compromise**

ABE PERKS

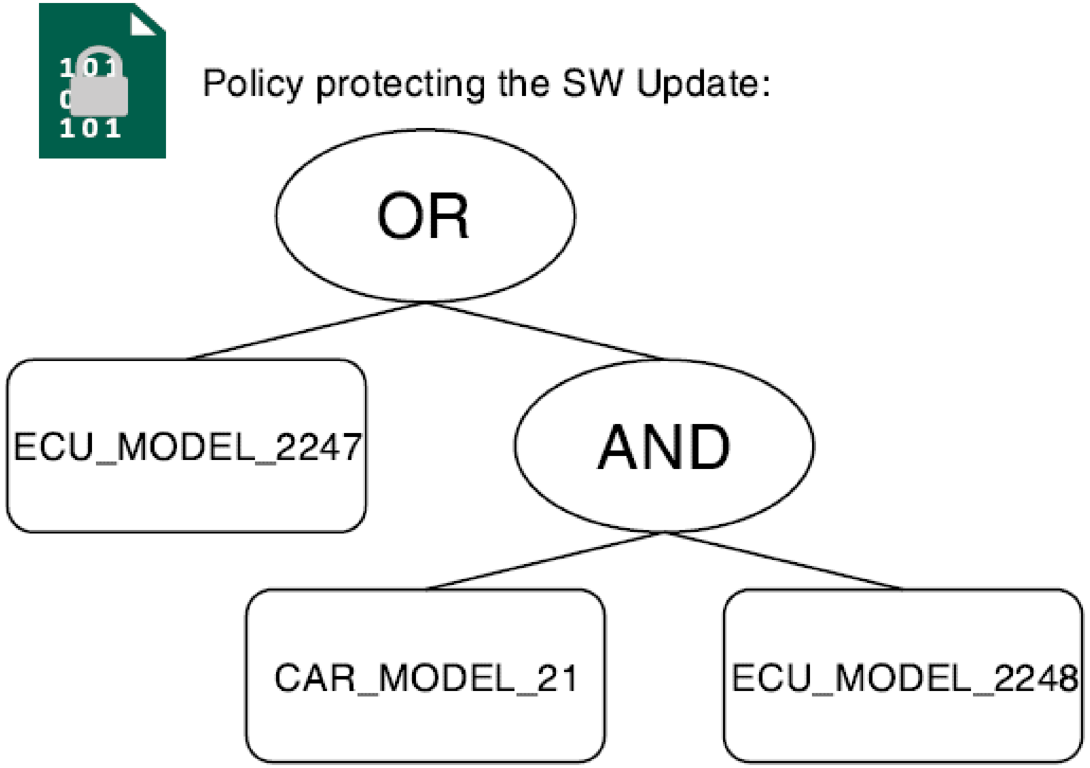
- Single Encryption – Multiple Decryption → High Scalability
- Fine-Grainedness → High Flexibility
- Mathematically Enforced Access Control Mechanism → Cryptographically Secure!
- Only one Public Key in the whole system.
- Users with identical Attribute Set (or Policy) have two different Decryption Keys.
- Some carefully engineered schemes are feasible on constrained IoT devices
- Several schemes in the literature to optimize different requirements: bandwidth, energy consumption, CPU efficiency…

REALISTIC ABE USE-CASES

EXAMPLE CP-ABE USE-CASE SCENARIO

Automotive ECUs' Over-The-Air SW Update

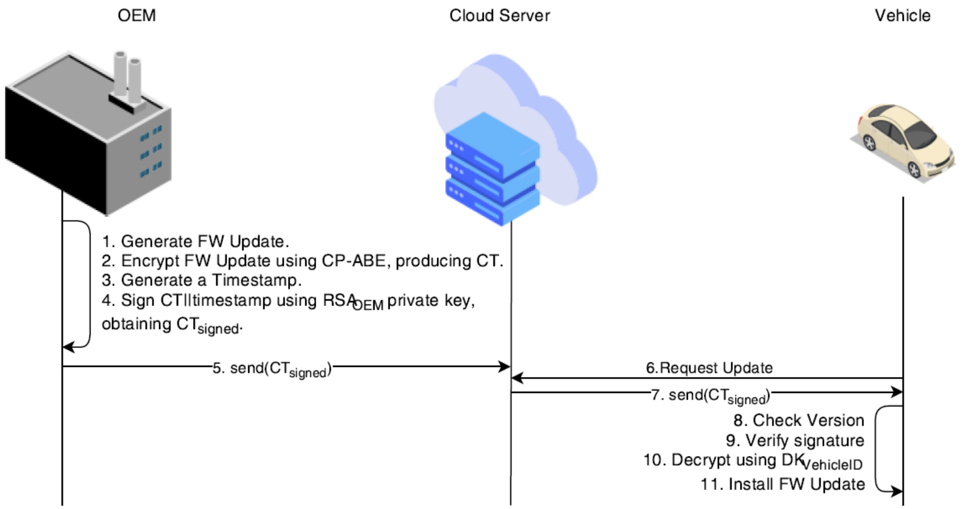
Policy protecting the SW Update:

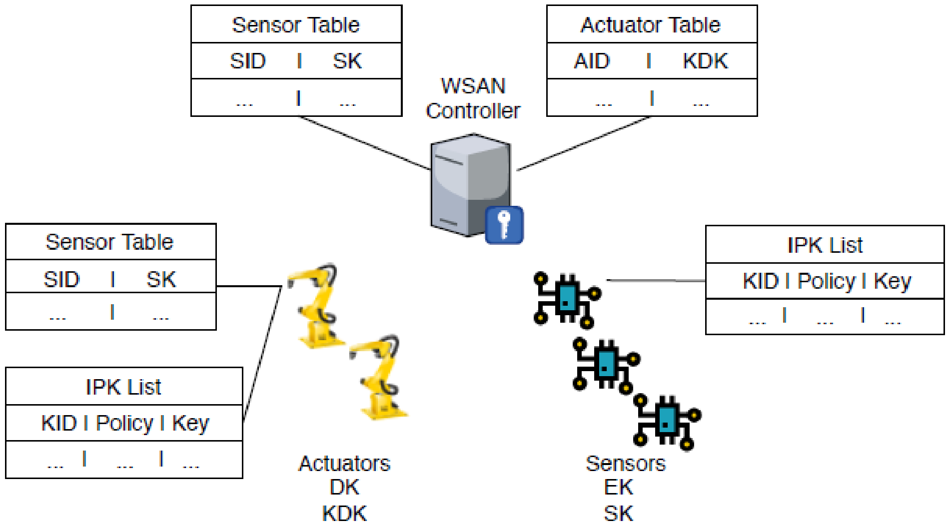


Vehicle1's attribute set:
{ CAR_MODEL_23;
ECU_MODEL_2247;
ECU_MODEL_2256;
ECU_MODEL_2268; }

Vehicle2's attribute set:
{ CAR_MODEL_21;
ECU_MODEL_2246;
ECU_MODEL_2248;}

ARE THERE ANY REALISTIC ABE IN IOT USE CASE?

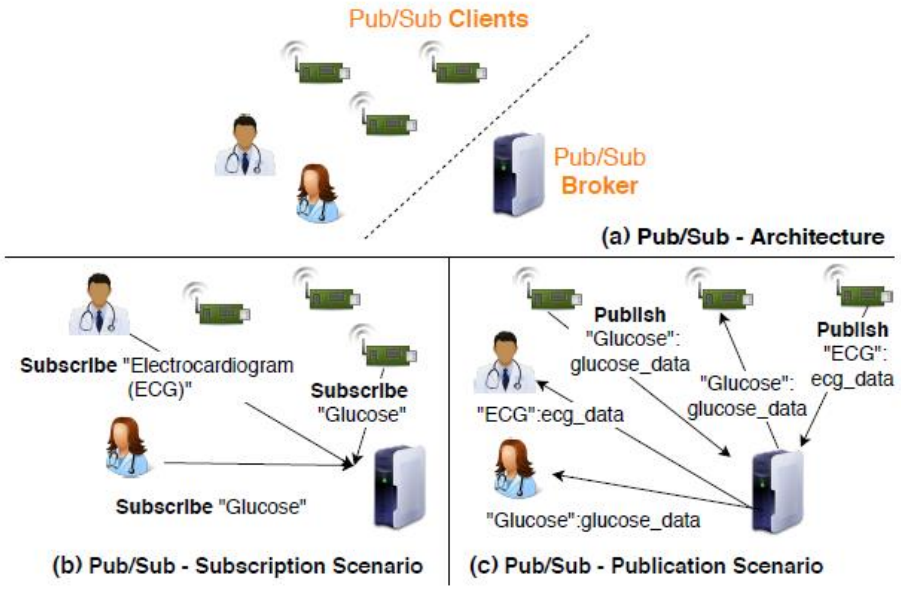


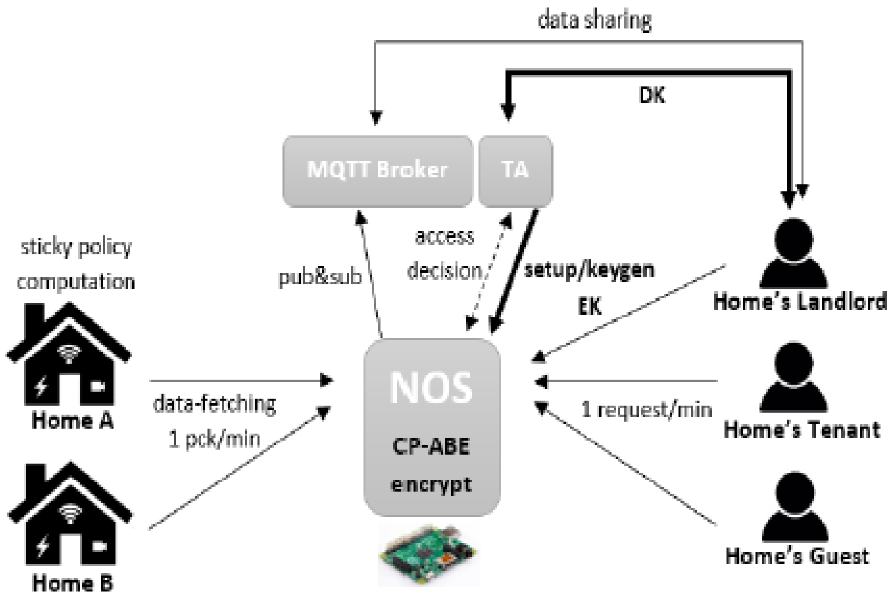


Automotive Scenario:
"Performance evaluation of attribute-based encryption in automotive embedded platform for secure software over-the-air update", MDPI Sensors

Industrial IoT Scenario:
"fABElous: An attribute-based scheme for industrial internet of things" IEEE BITS 2019, part of IEEE SMARTCOMP 2019.

ARE THERE ANY REALISTIC ABE IN IOT USE CASE?

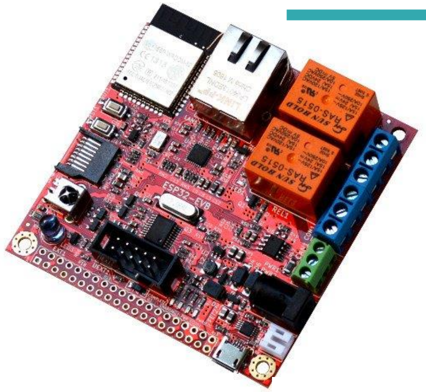




Healthcare Scenario:
"Performance evaluation of attribute-based encryption on constrained IoT devices"
Computer Communications.

Smart-Home Scenario:
"Attribute-based encryption and sticky policies for data access control in a smart home scenario: A comparison on networked smart object middleware"
International Journal of Information Security.

ORIGINAL CP-ABE PERFORMANCE IN IOT CONSTRAINED DEVICES

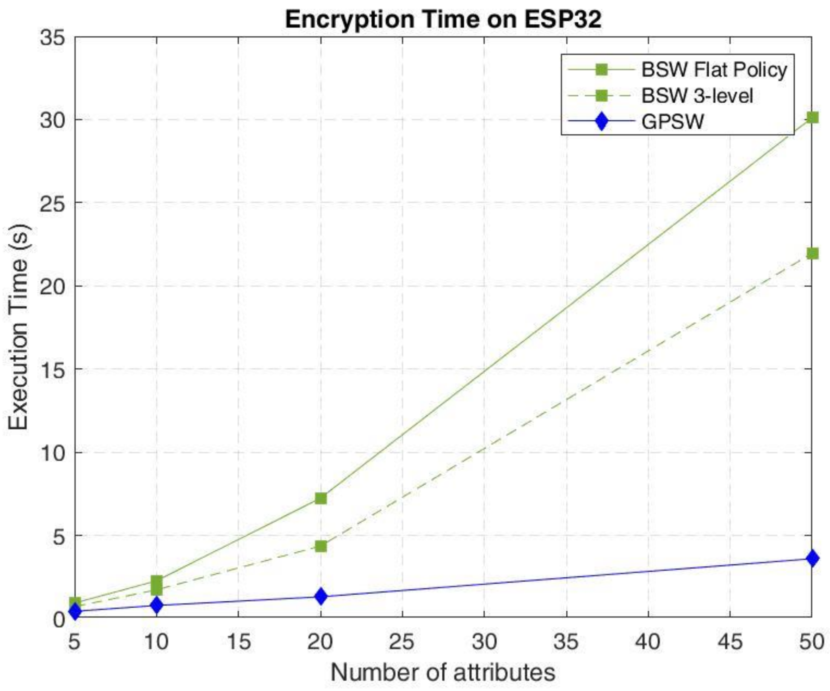


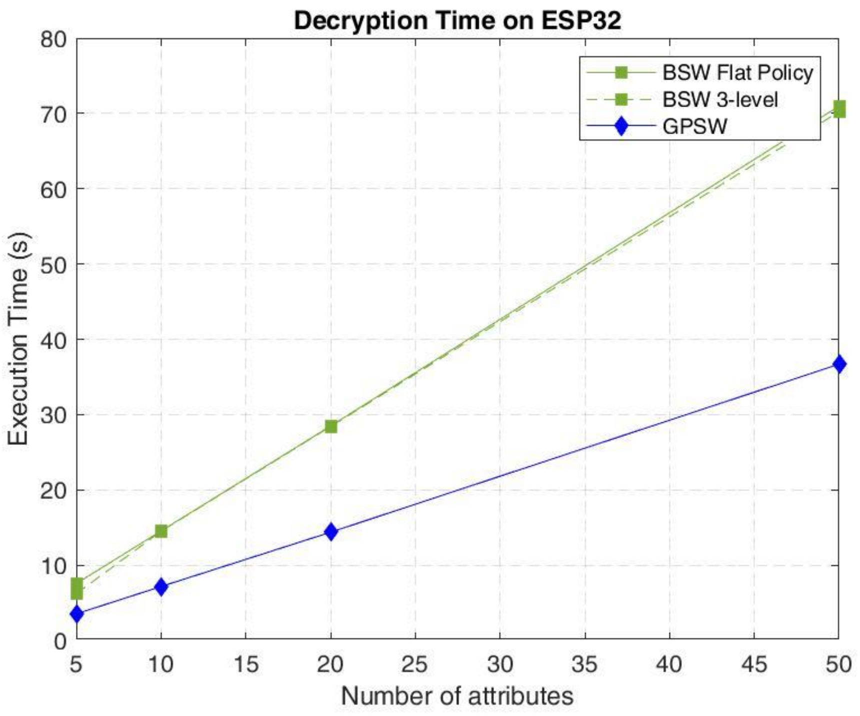
ESP32 Technical details:

CPU: Tensilica Xtensa dual-core LX6 microprocessor, operating at 240 MHz

Radio: Wi-Fi: 802.11 b/g/n

Memory: 448 KB flash and 520 KB RAM





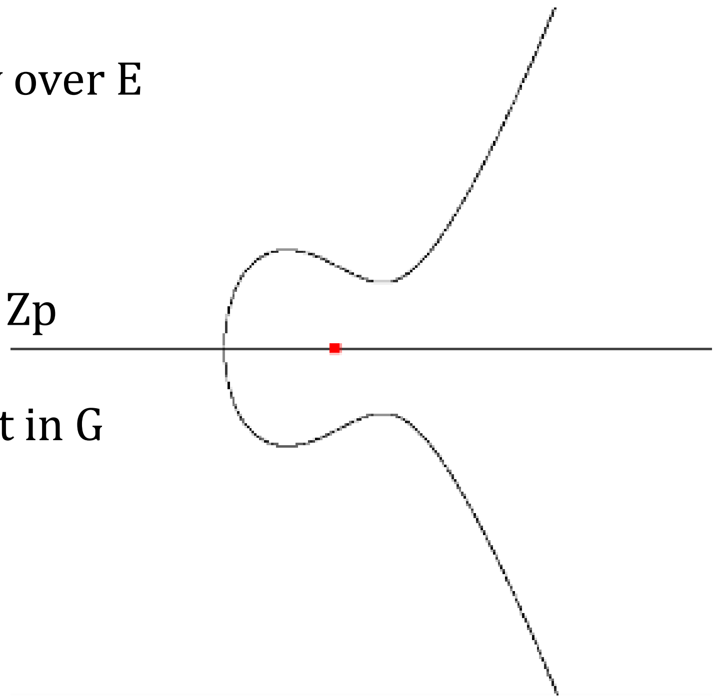
MATHEMATICAL FOUNDATIONS

ELLIPTIC CURVE CRYPTOGRAPHY

ABE can be implemented using different mathematics.
However, the most prominent constructions leverage ECC.

Some notation useful for later:

- ***p:*** a large prime which is also the base field size
- ***E:*** an elliptic curve defined over the prime field Fp
- ***G:*** a multiplicative cyclic group in which each element lay over E
- **GT**: a multiplicative cyclic group of integers of order p
- ***g:*** a generator of G
- ***Zp:*** a group of positive integer in [0,p-1]
- ***x^a:*** exponentiation operation. x belongs to G, a belongs to Zp
- ***xy:*** multiplication operation. Both x and y belong to G
- ***H(j)*** : j={0,1}* (sequence of bits), the output is an element in G



HARDNESS

Base Problem: Elliptic Curve Discrete Logarithm Problem (ECDLP)

Given two points "P,Q" belonging to G, find the integer "a" in Zp such that
$Q=P^a$

Best known algorithm: Parallelised Pollard's rho Algorithm (complexity $O\sqrt{p}$)

Parametrization: In order for ECC-based schemes (such as ABE) to be secure, p must be on enough bits to make the best-known attack unfeasible.

Log2(p)>= 2*security_level

Example: if 128 bits of security are needed, p should be a 256-bit prime number.

PAIRING-BASED CRYPTOGRAPHY

Given a multiplicative cyclic group G of prime order p and a target multiplicative cyclic group of the same order GT, a bilinear pairing is defined as

$Y = e(P,Q) \mid G \times G \rightarrow GT$

Is a function that maps two elements P and Q in G to an element of group GT fulfilling the following properties:

Bilinearity: for all a,b in Zp, $e(P^a, Q^b) = e(P, Q)^{ab} = e(P^b, Q^a)$

Non-degenerativity: $e(P,Q) != 1$

Efficient-computability: It exists an efficient algorithm to compute $e(P,Q)$

HARDNESS

Base Problem: Bilinear Diffie-Hellman Problem (BDHP)

Given three points P^a, P^b, P^c , find $e(P,P)^{abc}$

Best known algorithm: bruteforce attack

Parametrization: In order for PBC-based schemes (such as ABE) to be secure, p must be on enough bits to make the best-known attack unfeasible.

Log (p)>= security_level

Example: if 128 bits of security are needed, p should be a 128-bit prime number.

However, the "harder" constraint should be considered.
Therefore, p should be on 256 bits if 128 security bits are desired.

HARDNESS

Base Problem: Finite Field Discrete Logarithm Problem (FFDLP)

Given three points Y, W belonging to GT , find x | $Y=W^x$

Best known algorithm: Extended Tower Number Field Sieve (exTNFS)

Parametrization: In order for PBC-based schemes (such as ABE) to be secure, the size of GT elements must be on enough bits to make the best-known attack unfeasible.

Example: if 128 bits of security are needed, GT elements should be on ~5000 bits

This does not impact p. The order of GT remains "p", however the size of the elements must be increased. This means that there are only 2^{256} elements in GT and $2^{(5000-256)}$ other bit configurations are meaningless.

IN DEPTH: THE ORIGINAL CP-ABE PRIMITIVES

In the following slides, we will take a look at the primitives as described in the paper: "*Ciphertext-Policy Attribute-Based Encryption*" by Bethencourt, Sahai, and Waters.
https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4223236

Setup. The setup algorithm will choose a bilinear group $\mathbb{G}_0$ of prime order p with generator g. Next it will choose two random exponents $\alpha, \beta \in \mathbb{Z}_p$. The public key is published as:

$$\mathrm{PK} = \mathbb{G}_0, g, h = g^{\beta}, f = g^{1/\beta}, e(g,g)^{\alpha}$$

and the master key MK is (β, g^{α}).

IN DEPTH: THE ORIGINAL CP-ABE PRIMITIVES

KeyGen(MK, S). The key generation algorithm will take as input a set of attributes S and output a key that identifies with that set. The algorithm first chooses a random $r \in \mathbb{Z}_p$, and then random $r_j \in \mathbb{Z}_p$ for each attribute $j \in S$. Then it computes the key as

$$\mathrm{SK} = \big(D = g^{(\alpha+r)/\beta},$$
$$\forall j \in S: \quad D_j = g^r \cdot H(j)^{r_j}, D'_j = g^{r_j}\big).$$

ATTRIBUTE SET

-Physician
-Pisa Hospital
-Cardiology
-Professor
-EU Project "Health"

KEY

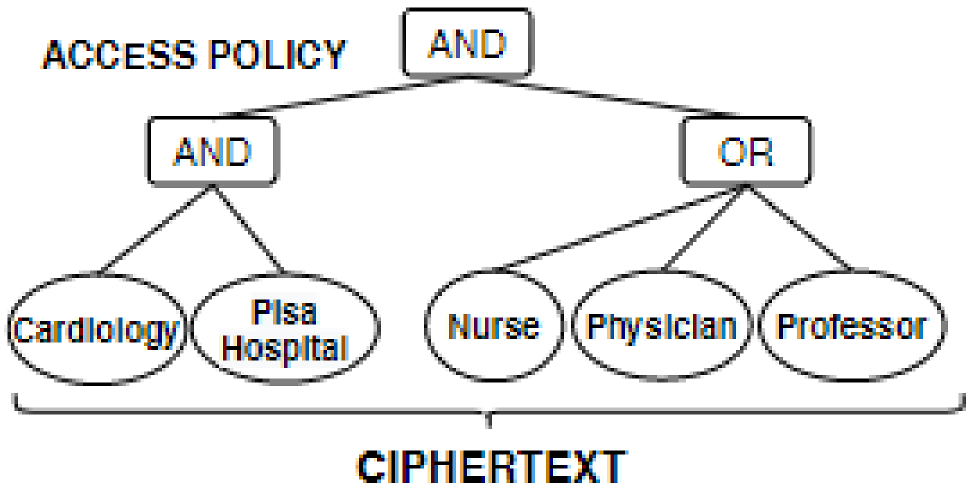
IN DEPTH: THE ORIGINAL CP-ABE PRIMITIVES

Encrypt(PK, M, $\mathcal{T}$). The encryption algorithm encrypts a message M under the tree access structure $\mathcal{T}$. The algorithm first chooses 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, $d_x = k_x - 1$.

Starting with the root node R the algorithm chooses a random $s \in \mathbb{Z}_p$ and sets $q_R(0) = s$. Then, it chooses d_R other points of the polynomial q_R randomly to define it completely. For any other node x, it sets $q_x(0) = q_{\mathsf{parent}(x)}(\mathsf{index}(x))$ and chooses d_x other points randomly to completely define q_x.

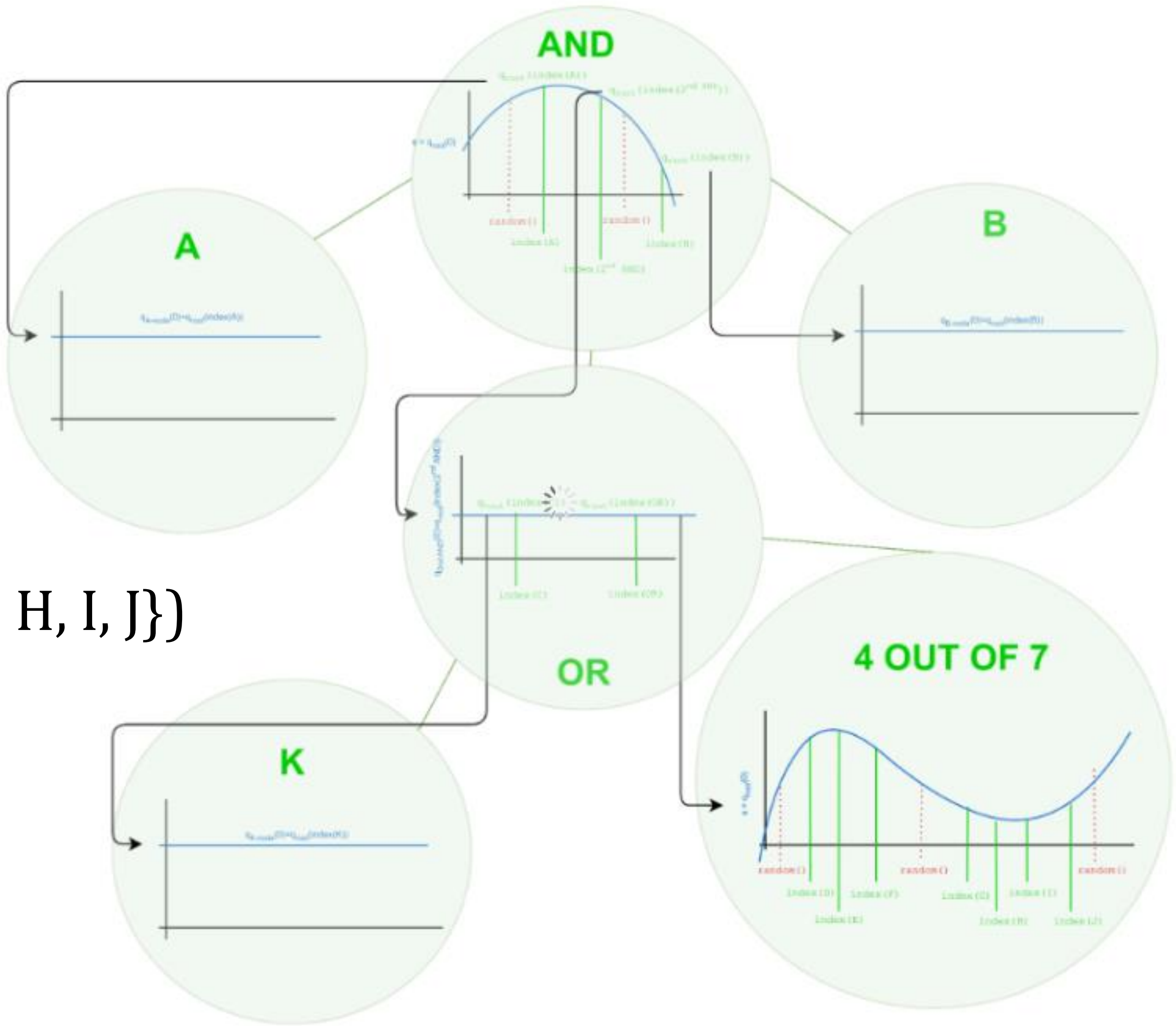
Let, Y be the set of leaf nodes in $\mathcal{T}$. The ciphertext is then constructed by giving the tree access structure $\mathcal{T}$ and computing

$$\mathrm{CT} = \big(\mathcal{T},\ \tilde{C} = Me(g,g)^{\alpha s},\ C = h^s,$$
$$\forall y \in Y: \quad C_y = g^{q_y(0)}, C'_y = H(\mathsf{att}(y))^{q_y(0)}\big).$$



ENCRYPTION EXAMPLE

A AND B AND
(K OR (4 OUT OF {D, E, F, G, H, I, J})



IN DEPTH: THE ORIGINAL CP-ABE PRIMITIVES

Decrypt(CT, SK). 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 DecryptNode(CT, SK, x) that takes as input a ciphertext CT $= (\mathcal{T}, \tilde{C}, C, \forall y \in Y: C_y, C'_y)$, a private key SK, which is associated with a set S of attributes, and a node x from $\mathcal{T}$.

If the node x is a leaf node then we let $i = \mathsf{att}(x)$ and define as follows: If $i \in S$, then

$$\begin{aligned}
\text{DecryptNode}(\text{CT}, \text{SK}, x) &= \frac{e(D_i, C_x)}{e(D'_i, C'_x)} \\
&= \frac{e\left(g^r \cdot H(i)^{r_i}, h^{q_x(0)}\right)}{e(g^{r_i}, H(i)^{q_x(0)})} \\
&= e(g,g)^{r q_x(0)}.
\end{aligned}$$

If $i \notin S$, then we define DecryptNode(CT, SK, x) $= \perp$.

We now consider the recursive case when x is a non-leaf node. The algorithm DecryptNode(CT, SK, x) then proceeds as follows: For all nodes z that are children of x, it calls DecryptNode(CT, SK, 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 $\perp$.

Otherwise, we compute

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

and return the result.

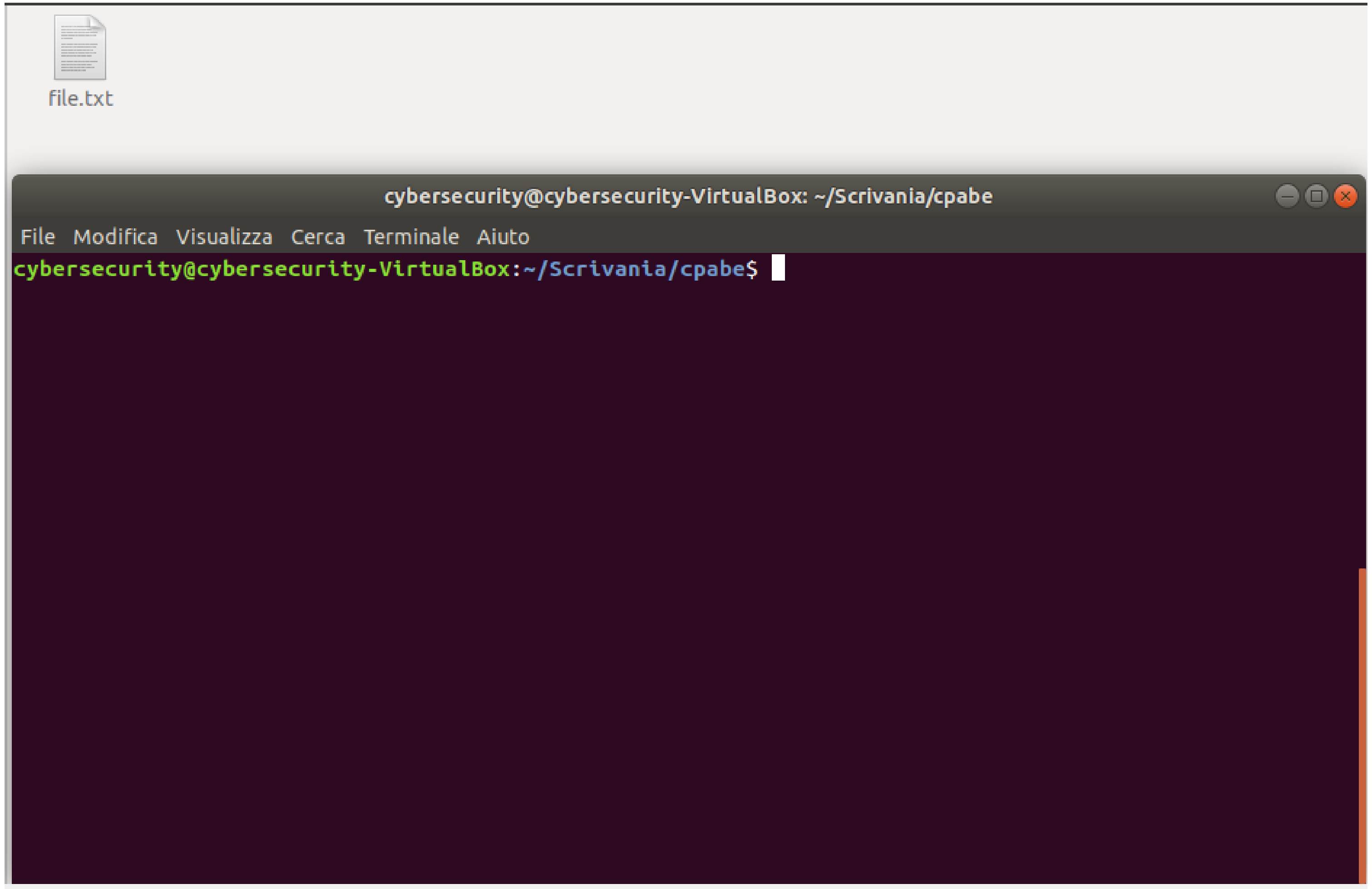
Now that we have defined our function DecryptNode, we can define the decryption algorithm. The algorithm begins by simply calling the function on the root node R of the tree $\mathcal{T}$. If the tree is satisfied by S we set $A =$ DecryptNode(CT, SK, r) $= e(g,g)^{r q_R(0)} = e(g,g)^{rs}$. The algorithm now decrypts by computing

$$\tilde{C}/(e(C,D)/A) = \tilde{C}/\left(e\left(h^s, g^{(\alpha+r)/\beta}\right)/e(g,g)^{rs}\right) = M.$$

CP-ABE TOOLKIT

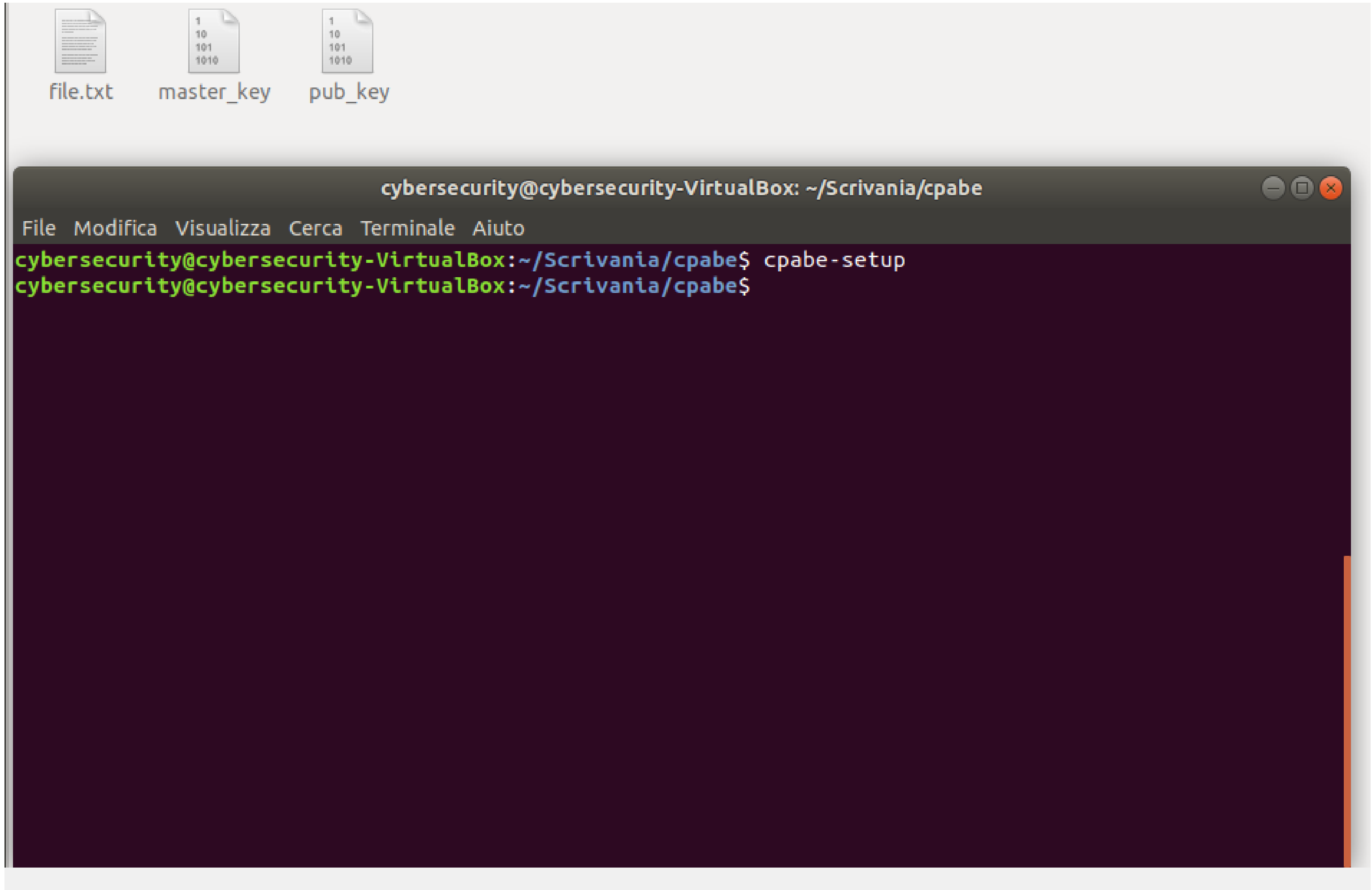
CP-ABE TOOLKIT

In a dedicate folder I created a file to be encrypted and opened the terminal



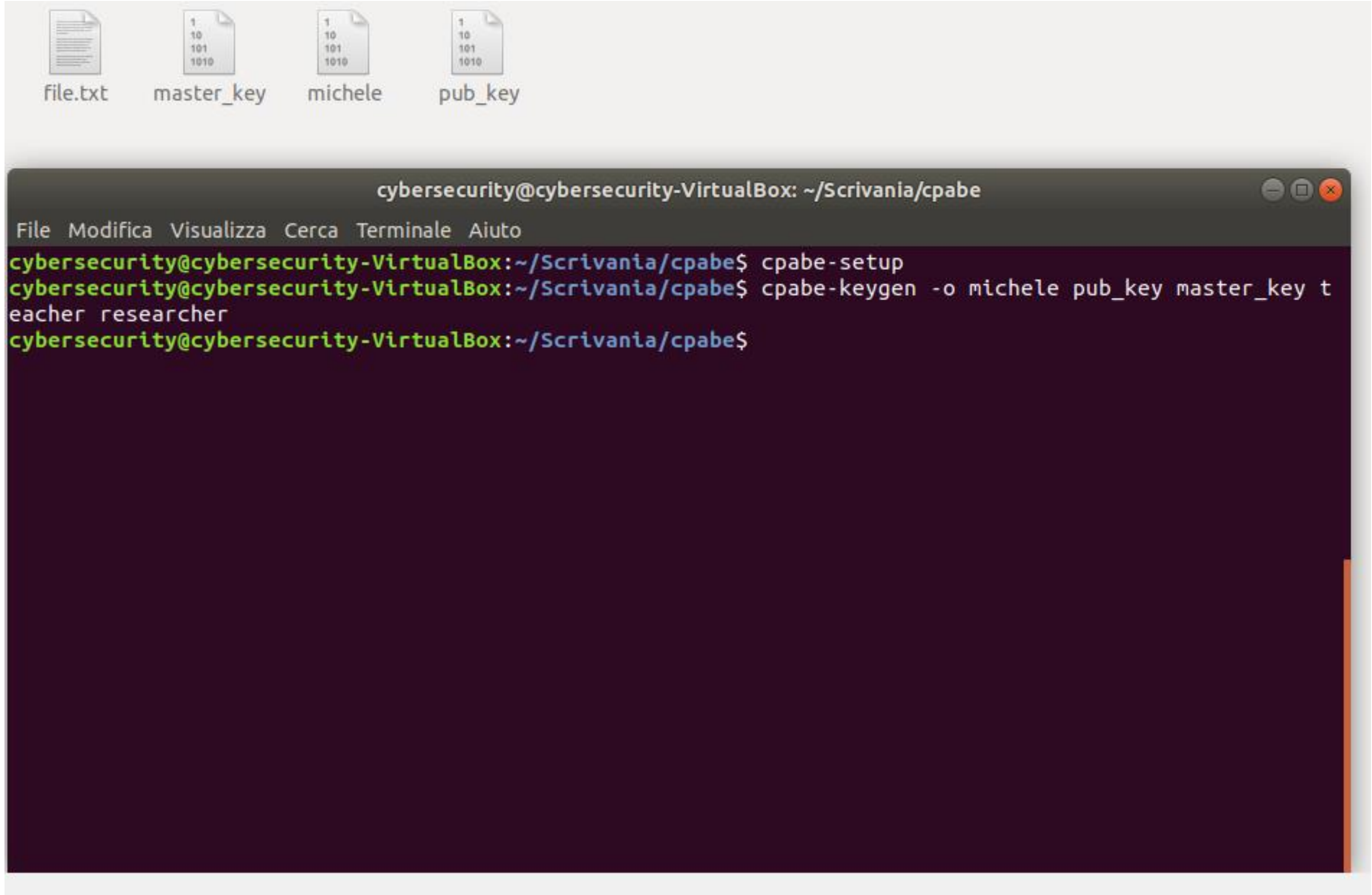
CP-ABE SETUP

Running cpabe-setup creates the public key (useful for encryption and decryption) and the master key (useful for decryption).



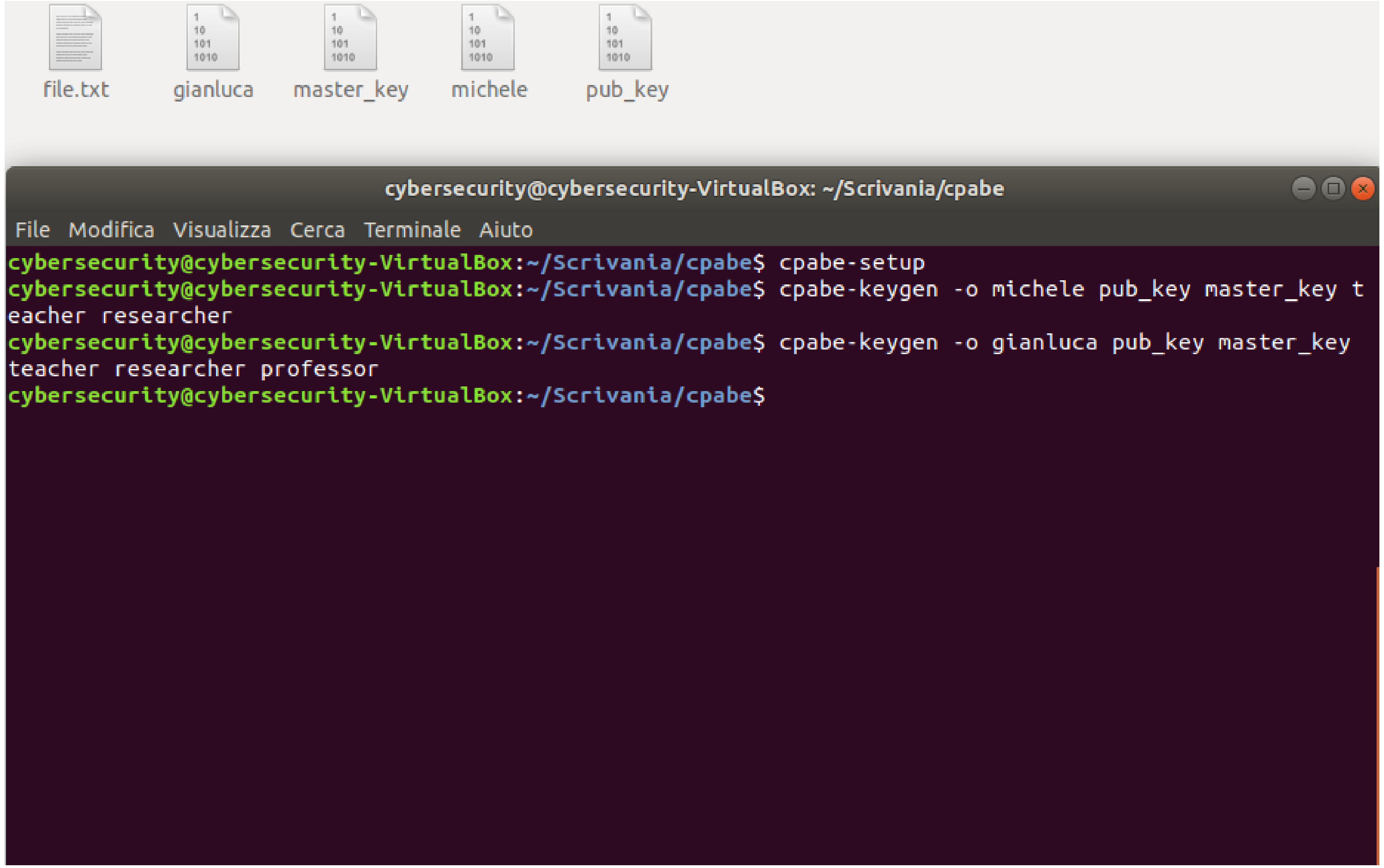
CP-ABE KEYGEN

cpabe-keygen [OPTION ...] PUB_KEY MASTER_KEY ATTR [ATTR ...]



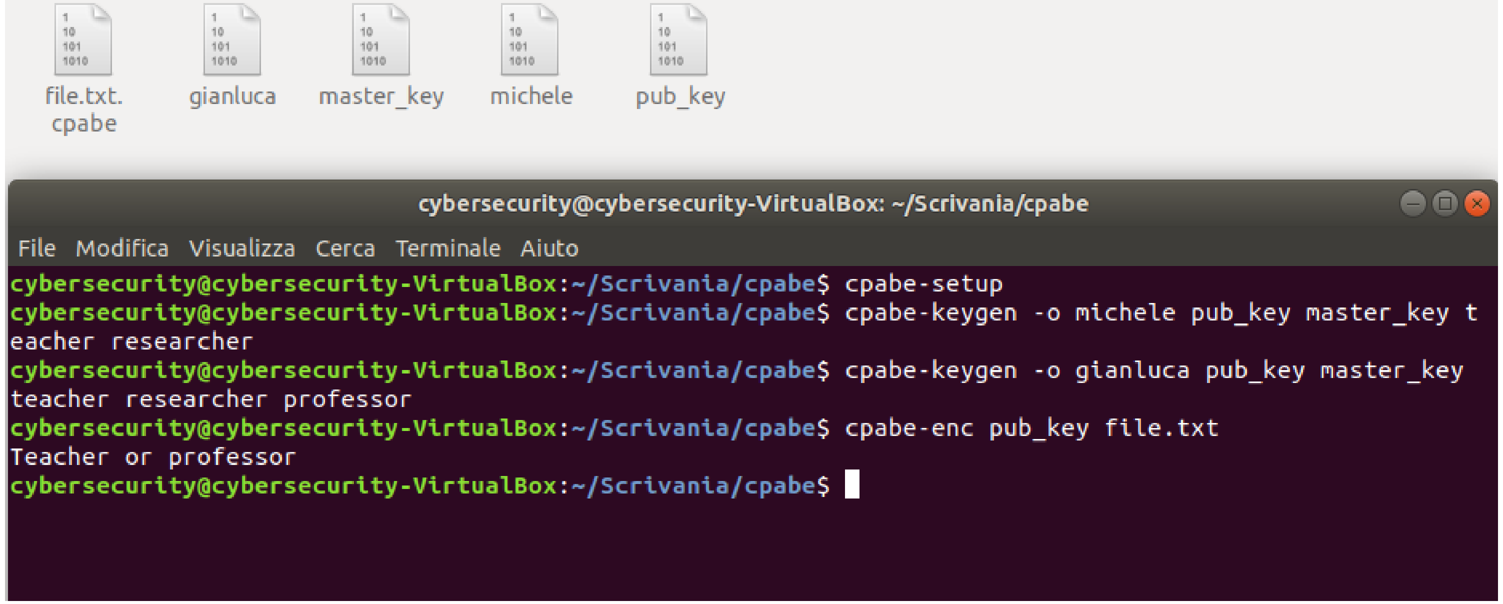
CP-ABE KEYGEN PT2

cpabe-keygen [OPTION ...] PUB_KEY MASTER_KEY ATTR [ATTR ...]



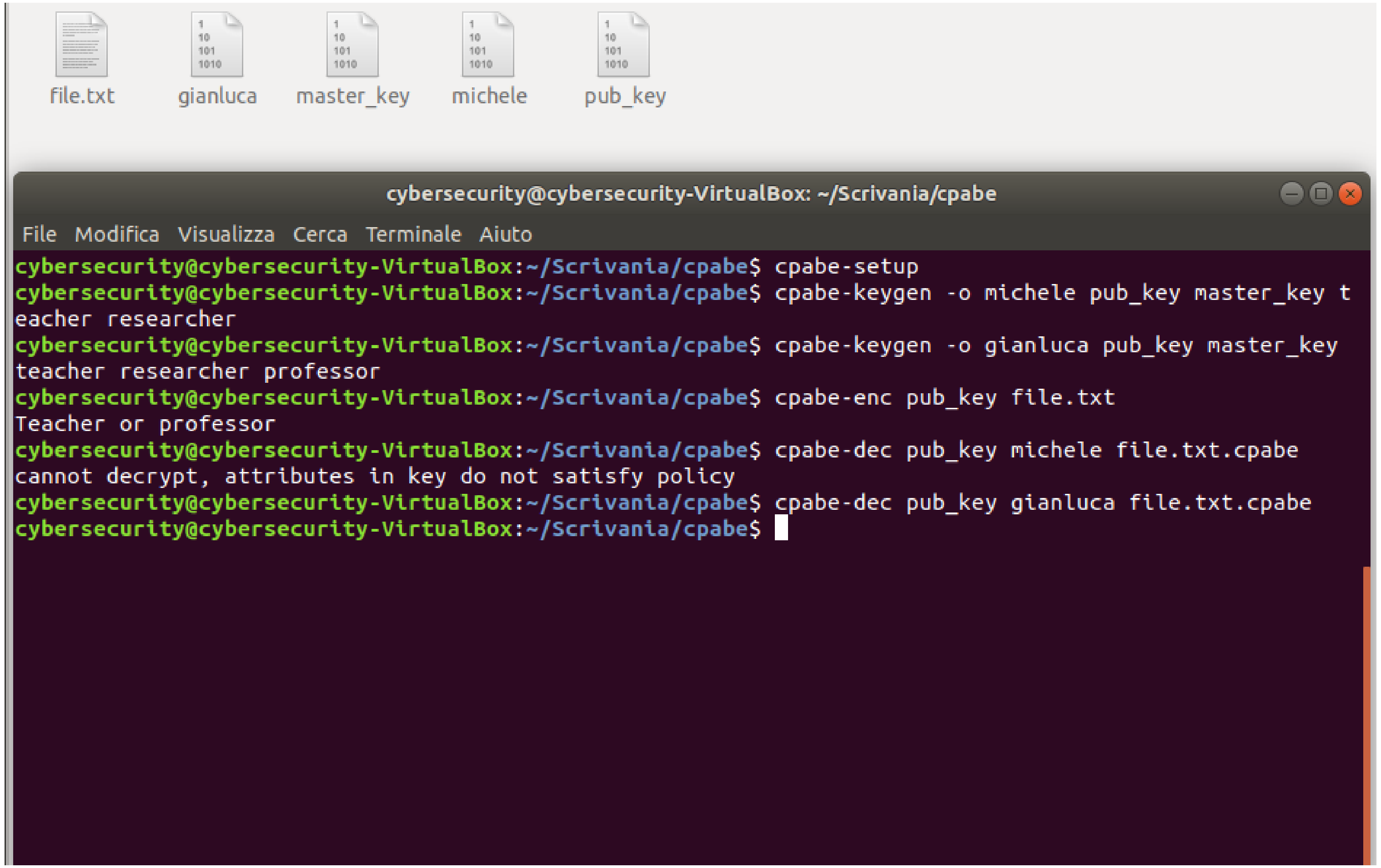
CP-ABE ENCRYPT

cpabe-enc [*OPTION* ...] *PUB_KEY FILE* <enter> [*POLICY*] <ctrl-d to encrypt>



CP-ABE DECRYPT

cpabe-dec [*OPTION* ...] *PUB_KEY PRIV_KEY FILE*



HANDS-ON

MEET LOODLE!

Loodle is a big tech corporation based in Italy.
Loodle offers web services as well as electric products built in its factories.
In the following, we list some loodle's employees and some files that loodle produces.

Alice: Loodle CEO. She has access to every document produced inside her company.

Beatrice: Loodle COO. She has access to the every document regarding collaborations with other companies. Moreover, she has access to the documents about loodle's stock market.

Claudio: One of the Loodle's higher managers. He has access to every document produced by the sales and marketing department.

Dario: One of the Loodle's higher managers. He has access to every document produced by the production line. He is in the R&D and he also has access to every accident report filed by Loodle's engineers.

MEET LOODLE!

Elia: is an higher manager in the marketing dept. He has access to the new products catalogue and client’s data in order to create the marketing campaign of the next season.

Francesca: is the Senior Cybersecurity Engineer. She coordinates the defense of loodle’s network and has access to every document produced by the cybersecurity department. Moreover, she has access to every failure report of any kind in Loodle.

Giulio: is a loodle’s analyst. He has access to client’s data and loodle’s finantial reports.

Irene: SW developer in the R&D department of loodle. She has access to every prototype projects and on any past projects as well as complaints from users.

LOODLE'S FILES

Stock Market report, 1Q 2022: Contains finantial report about the stock market from Jan 1 2022 to Mar 31 2022.

TARA on Loodle's Internal Network: Threat Analysis and Risk Assessment on the management of the internal loodle's network. Performed by the cybersecurity dept.

Assembly line Accident report: An employee was injured in the assembly line. In the report there are the details of the accidents.

Joint Venture proposal w/ Pear Inc.: A formal contract proposal with the company Pear to launch a new service for Pear and Loodle users.

Selling report period 2020/2021: Detailed report on quantities, prices, and costs of each loodle's product and services.

Complaints on Loodle's product: List of users complaints on loodle's product collected from various sources: social networks, forms on loodle's website, loodle store, etc… File produced by marketing department.

LOODLE’S FILES

Report on the effectiveness of the marketing campaign: Polling-collected information about the receptiveness of loodles latest marketing campaign. Available only to higher managers related to marketing.

Prototype Projects #42: Technical projects (schemes, diagrams, drawings, sw) of a potentially new loodle’s product.

Technical Details of the production line machinery: Manuals and description of the machineries that loodle bought for its factories.

Report on the failure of the autonomous kart in loodle’s warehouse: Autonomous kart bumped in each other a couple of month ago for an entire day. This is the report of the failure.

Clients data on age, country, and commonly taken paths: data that loodle’s users consented to share with loodle.

ASSIGNMENT

Starting from the descryption of data and employee, define:

Access Policies: for each data to be protected. Remember! The access policy answers to the question “Who can access this data?” or better “People with which attributes can access this data?”

Attribute Sets: for each employee. Remember! The attribute set answers to the question “How can I describe this employee?” or better “What attribute does this employee have?”.