De Cifris Trends in *Cryptographic Protocols*

University of Trento and *De Componendis Cifris*

October 2023

Lecture 8

Barbara Masucci **University of Salerno**

Hierarchical Key Assignment

Barbara Masucci

University of Salerno

OUTLINE OF THE TALK

- The Access Control Problem
  -Motivations, scenario, requirements

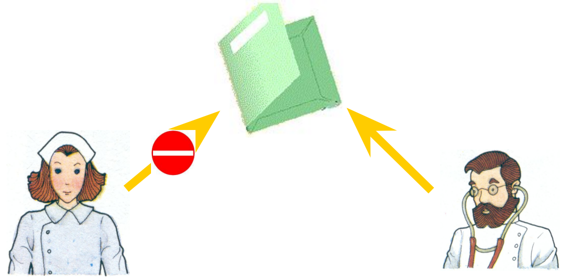
- Hierarchical Key Assignment Schemes (HKAS)
  -Definition, evaluation criteria, notions of security

- Provably secure constructions for HKAS
  -PRF-based, EBC-based

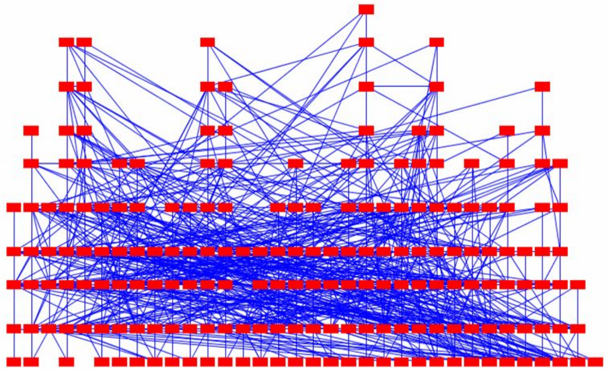
- Some extensions:
  -Time-bound HKAS, HKAS supporting dynamic updates,
  hierarhical and shared access control

THE ACCESS CONTROL PROBLEM: MOTIVATIONS

- Only authorized users should be given access to sensitive resources

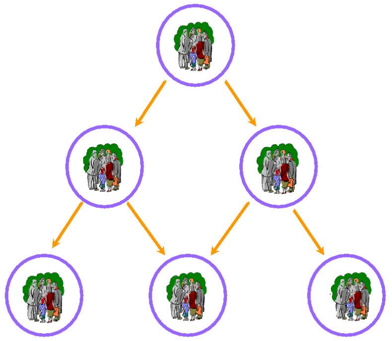


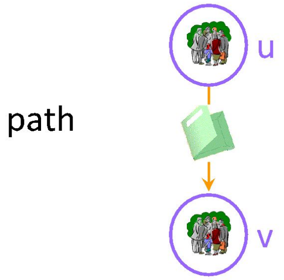
- Many environments are characterized by a hierarchical structure
    - Healthcare
    - Military and Government
    - Databases
    - Broadcast services
    - Networking



HIERARCHICAL ACCESS CONTROL: THE SCENARIO

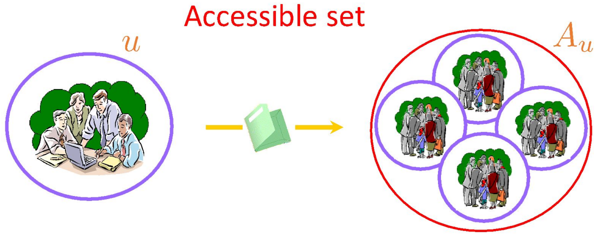
- According to their competencies, roles and responsibilities, users are organized in a hierarchy formed by disjoint security classes
- A partial order relation is defined according to authority, position or power of classes
- A partially ordered hierarchy can be represented by a directed graph *G=(V,E)*



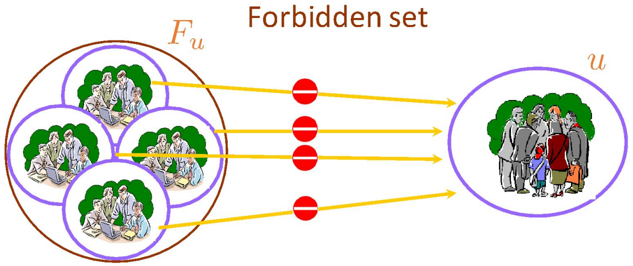


THE ACCESS CONTROL PROBLEM: REQUIREMENTS

- Any class should be able to access secret data of all its successors in the hierarchy



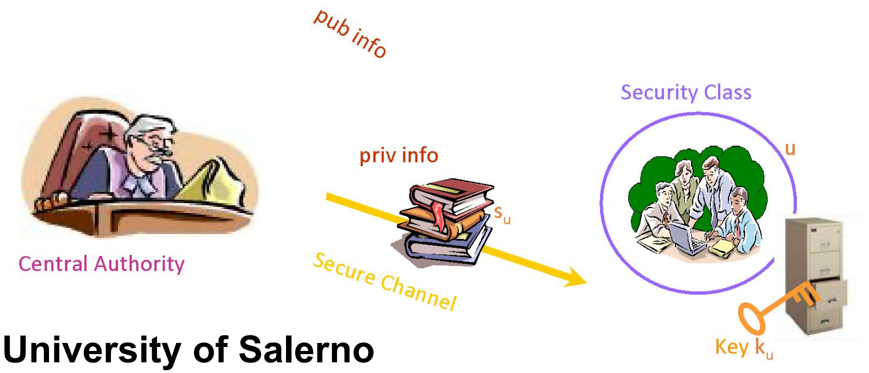
- Any set of classes should not be able to access secret data of a class which is not a successor of any class in the set



THE ACCESS CONTROL PROBLEM: CRYPTOGRAPHIC SOLUTION

- Implement hierarchical access control policies by means of Cryptography
  - Assign and manage the keys according to the policy

- *Hierarchical Key Assignment Schemes (HKAS)*
  - Assign an *encryption key* and some *private information* to each class in the hierarchy, as well as some *public information*



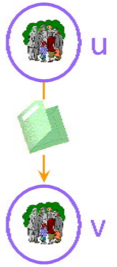
HIERARCHICAL KEY ASSIGNMENT: DEFINITION

A hierachical key assignment scheme for $G=(V,E)$ is a pair of algorithms (Gen,Der):

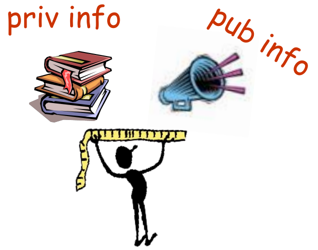
- $(s,k,pub) \longleftarrow Gen(1^{T}, G)$

  - s is the sequence of private information
  - k is the sequence of keys
  - pub is the sequence of public information

- $k_{v} \longleftarrow Der(1^{T}, G, u, v, s_{u}, pub)$ for each class v in A_u

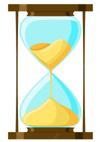


HIERARCHICAL KEY ASSIGNMENT: EVALUATION CRITERIA

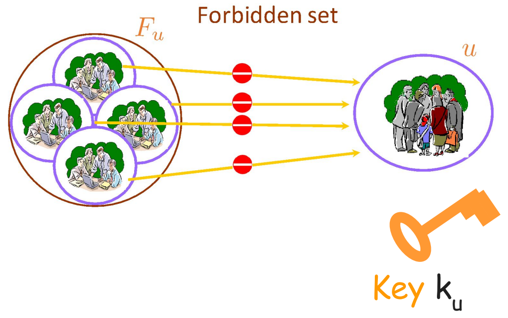


- Size of the information stored by each class
- Amount of public information
- Communication complexity of key updates
  - How much secret/public data needs to be re-distributed?

- Efficiency of key derivation



- Security against collusion attacks
  - Provable security (more on this later)



Barbara Masucci University of Salerno

THE AKL-TAYLOR SCHEME (1983)

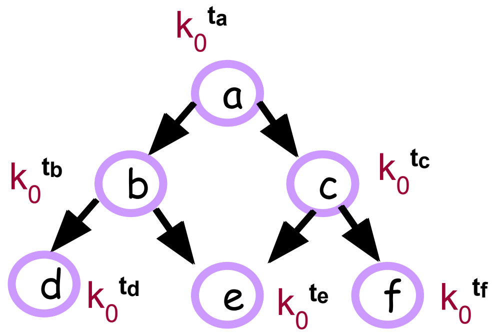
Algorithm *Gen(1^τ, G)*
- Choose two large primes p and q and compute $n=pq$
- Choose uniformly at random a secret $1<k_0<n$
- Assign to each class u
  - a public value t_u such that t_u divides t_v iff $v \in A_u$
  - the key $k_u = k_0^{t_u} \bmod n$
  - the private information $s_u = k_u$

Public values assignment
- Assign a prime p_u to each class u
- Compute the public value

$$t_u = \begin{cases} 1 & \text{if } A_u=V \\ \prod_{v \notin A_u} p_v & \text{otherwise} \end{cases}$$

Algorithm *Der(1^τ, G, u, v, s_u, pub)*
- Extract t_u and t_v from pub
- Compute the key k_v as follows:

$$k_v = (k_u)^{tv/tu} = (k_0)^{tv} \bmod n$$



$pub=(t_a,t_b,t_c,t_d,t_e,t_f)$

Barbara Masucci **University of Salerno**

THE AKL-TAYLOR SCHEME (1983)

- Pros
  - Low private storage
  - Direct key derivation
  - Moderate public storage

- Cons
  - Key derivation involves modular exponentiation with large exponents
  - Key update requires re-distribution of private information
  - Security of the scheme is based on the assumption that computing roots modulo a product of large primes is difficult
    - At the time of the proposal a formal model for hierarchical key assignment was missing

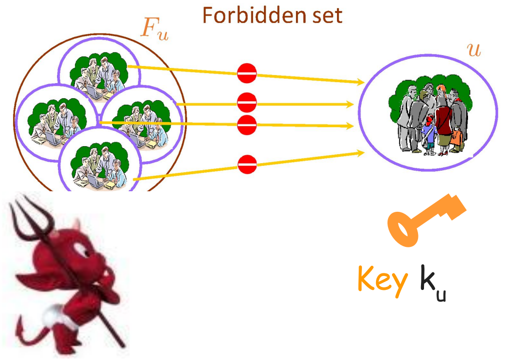
PROVABLE SECURITY UNDER A COMPLEXITY ASSUMPTION

- Several other schemes have been proposed in the last 40 years
  - Many of them lack a formal security proof and have been shown to be insecure

- In 1984 Goldwasser and Micali introduced the use of security reductions
  - Aim at reducing the security of a protocol to the difficulty of solving a (presumed) hard computational problem

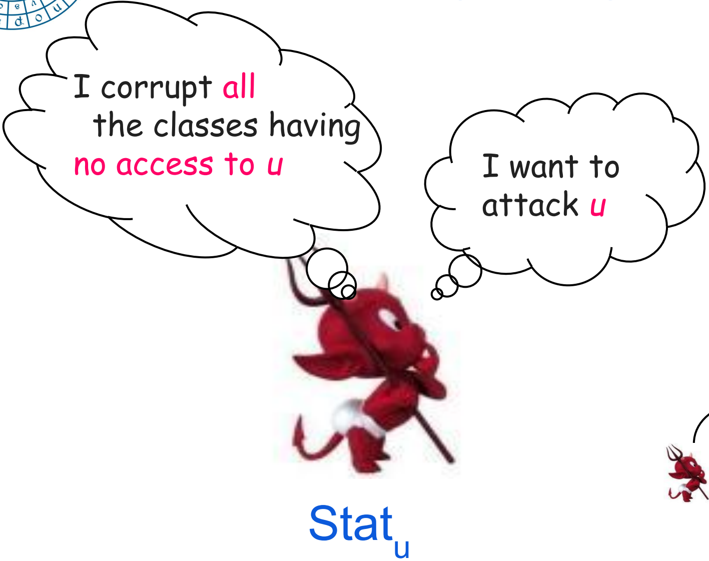
HIERARCHICAL KEY ASSIGNMENT: NOTIONS OF SECURITY

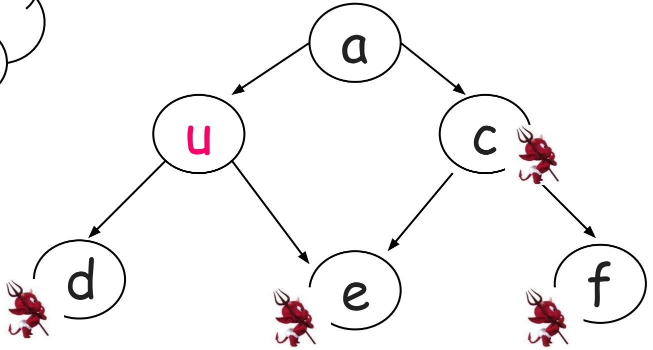
- Atallah et al. (2005) formally defined security for a HKAS by considering
  - *Adversarial behaviour*
    - *What can an adversary do?*
  - *Adversarial goal*
    - *Which game does the adversary play?*



Barbara Masucci University of Salerno

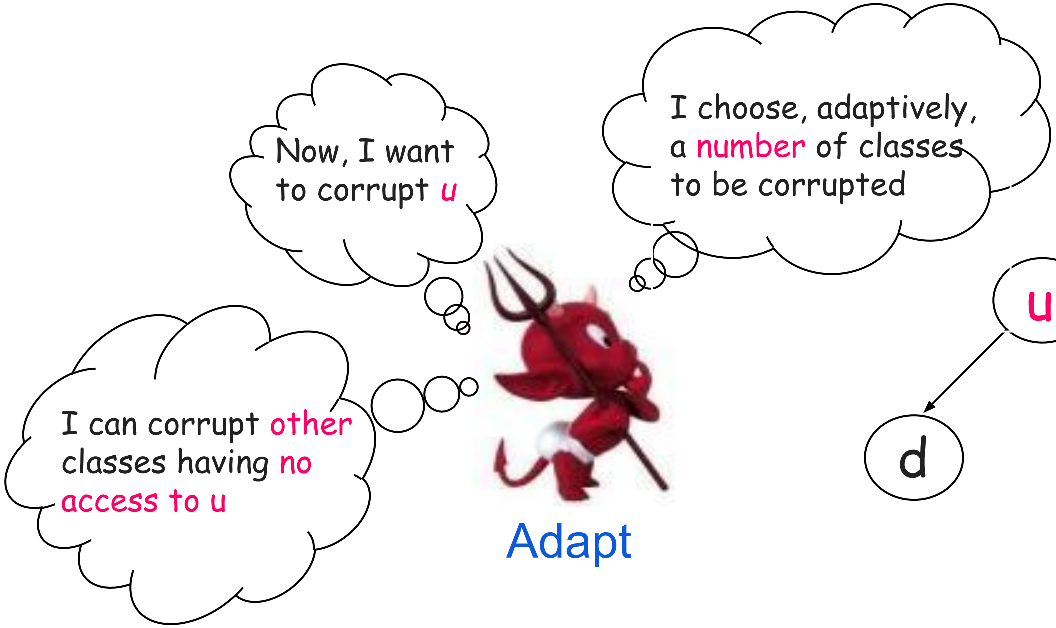
WHAT CAN AN ADVERSARY DO? STATIC ADVERSARY

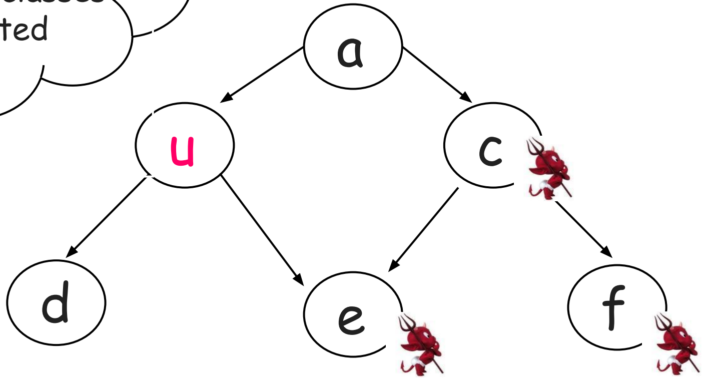




The adversary chooses to attack class *u* before the setup of the scheme

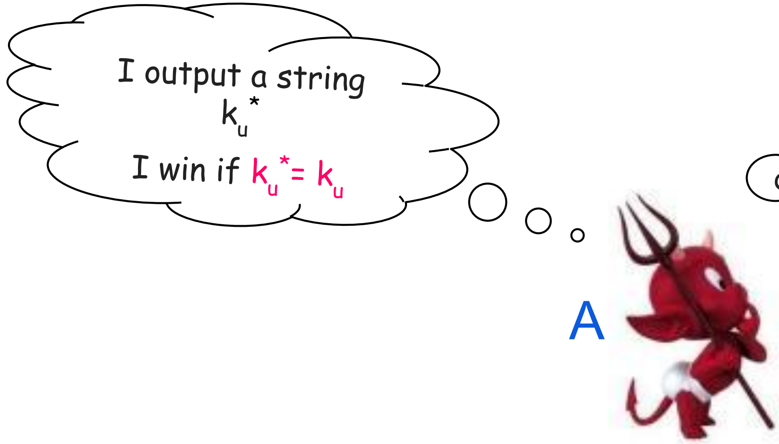
WHAT CAN AN ADVERSARY DO?
ADAPTIVE ADVERSARY


Now, I want to corrupt u
I choose, adaptively, a number of classes to be corrupted
I can corrupt other classes having no access to u
Adapt

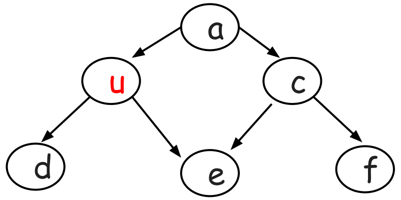


The adversary first gets all public information and private information of some classes and then chooses the class to attack

WHICH GAME DOES THE ADVERSARY PLAY? KEY RECOVERY



A

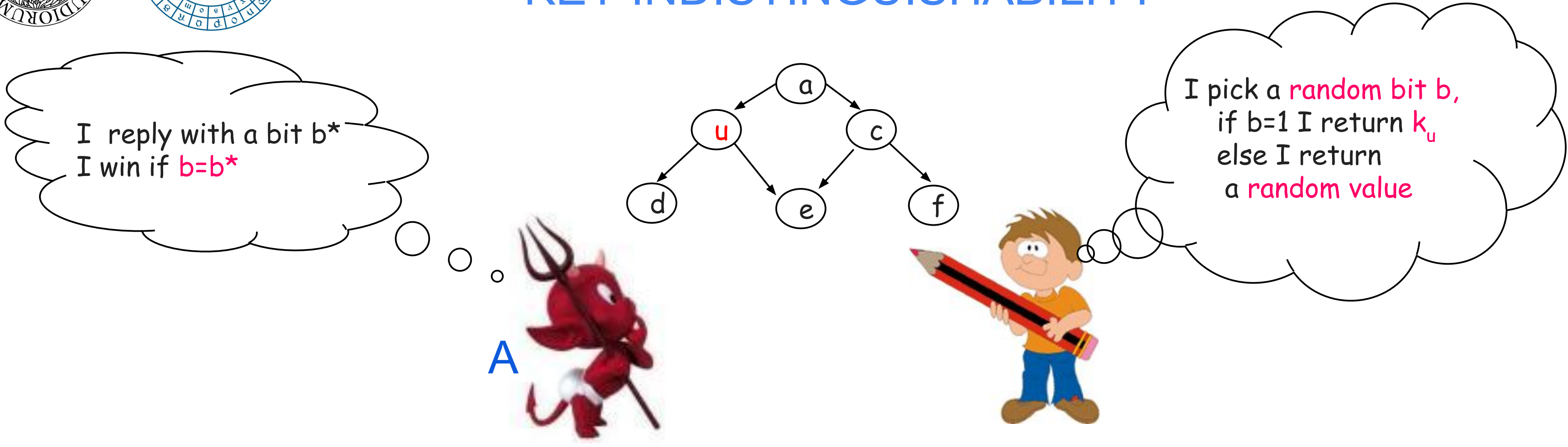


$$\mathrm{Adv}^{\mathrm{REC}}_{A} = \Pr[\, k_u^* = k_u]$$

The scheme is secure against key-recovery if $\mathrm{Adv}^{\mathrm{REC}}_{A}$ is negligible

REC-ST if A is static
REC-AD if A is adaptive

WHICH GAME DOES THE ADVERSARY PLAY?
KEY INDISTINGUISHABILITY


I reply with a bit b*
I win if b=b*
A
a
u
c
d
e
f
I pick a random bit b, if b=1 I return k_u else I return a random value

$$\mathrm{Adv}^{IND}_{A} = \left| \Pr[\, b \leftarrow A \,] - 1/2 \right|$$

The scheme is secure w.r.t. key-indistinguishability if Adv^{IND}_{A} is negligible

IND-ST if A is static
IND-AD if A is adaptive

IMPLICATIONS AND SEPARATIONS

$$\begin{tikzcd}
\boxed{\text{IND-ST}} \arrow[r, shift left] \arrow[d, shift right] & \boxed{\text{REC-ST}} \arrow[l, shift left, "/" description] \arrow[d, shift right] \\
\boxed{\text{IND-AD}} \arrow[u, shift right] & \boxed{\text{REC-AD}} \arrow[u, shift right]
\end{tikzcd}$$

- Static and Adaptive adversaries are polynomially equivalent
- There exists a scheme which is REC-ST secure but not IND-ST secure

THE PRF-BASED CONSTRUCTION

Algorithm *Gen(1^τ, G)*

- Let F be a PRF family
- Assign to each class u
  - -a public value $l_u \in \{0,1\}^\tau$
  - -a key $k_u \in \{0,1\}^\tau$
  - -the private information $s_u = k_u$
- Assign to each edge (u,v) a public value $p_{(u,v)} = F(k_u, l_v) \oplus k_v$

Algorithm *Der(1^τ, G, u, v, s_u, pub)*

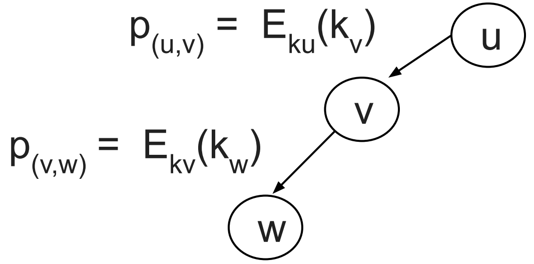
- Extract l_v and $p_{(u,v)}$ from pub
- If (u,v) є E, compute the key k_v as:

$$k_v = p_{(u,v)} \oplus F(k_u, l_v)$$

- Private storage: one key for each class
- Public storage: |E|+|V| values
- Key derivation: indirect
- REC-ST secure
- Not IND-ST secure

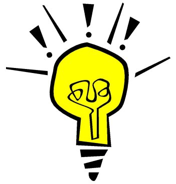
THE ENCRYPTION BASED CONSTRUCTION

- Let $\prod = (K, E, D)$ be a symmetric encryption scheme
- Encrypt the key k_v with the key k_u, for each edge (u,v)



- Private storage: one key for each class
- Public storage: |E| values
- Key derivation: indirect
- REC-ST secure under plaintext indistinguishability of the encryption scheme
- Not IND-ST secure

THE ENCRYPTION BASED CONSTRUCTION



- How to achieve IND-ST security?
  - Never use the key assigned to a class to encrypt the keys assigned to other classes!

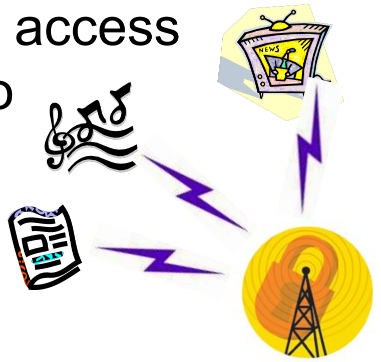
$$\begin{tikzcd}
 & p_{(u,v)} = \mathcal{E}_{s_u}(s_v) & u \arrow[dl] & p_{(u,u)} = \mathcal{E}_{s_u}(k_u) \\
p_{(v,w)} = \mathcal{E}_{s_v}(s_w) & v \arrow[dl] & p_{(v,v)} = \mathcal{E}_{s_v}(k_v) & \\
w & p_{(w,w)} = \mathcal{E}_{s_w}(k_w) & &
\end{tikzcd}$$

- Private storage: one key for each class
- Public storage: $|E| + |V|$ values
- Key derivation: indirect
- IND-ST secure under plaintext indistinguishability of the encryption scheme

Barbara Masucci University of Salerno

TIME-BOUND HKAS

- In several applications, users should access data only in specific periods of time
  - Examples: subscription services (digital libraries, music collections, newspapers, cable TV)
- A user may be assigned to a class for a certain time interval
- Once a time period expires, users should not be able to access any subsequent keys if they are not authorized to do so



HKAS SUPPORTING DYNAMIC UPDATES

- A HKAS supporting dynamic updates is a HKAS equipped with an updating algorithm Upd

$$(G', s', k', pub') \square \mathit{Upd}(1^{\tau}, G, s, k, pub, up)$$

-up: insertion/deletion of classes/edges, key replacement, user revocation

- The security model needs to address futher security issues introduced by Upd

  -A dynamic adaptive adversary can also perform dynamic updates on the hierarchy

HIERARCHICAL AND SHARED ACCESS CONTROL

- How to extend hierarchical access control in order to prevent abuses or violations?
  - -The Edward Snowden Case

- The access control should be not only hierarchical, but also shared
  - -Sensitive data should be accessed only by the agreement of some specific users (NSA Orange Book, Two-Person Authorization)

CONCLUSIONS

- Many environments are characterized by a hierarchical structure
- Access control in hierarchical structures can be implemented through HKASs
- We have analyzed different security models for HKAS, as well as some constructions
- We have also considered some extensions for such models

De Componendis Cifris

https:/www.decifris.it

Barbara Masucci University of Salerno