Modern Cryptography

Lecture 14, Advanced Encryption

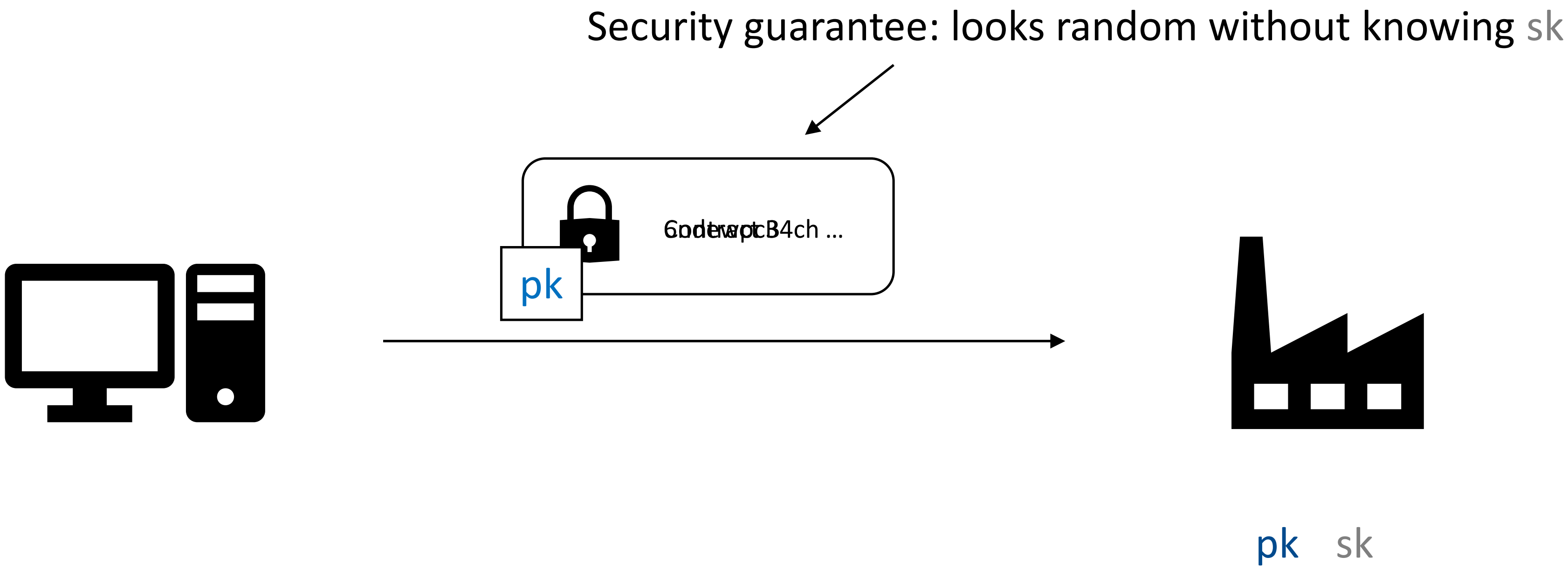
Christoph Striecks

Organizational

- Where to find the slides and homework?
  - https://danielslamanig.info/ModernCrypto18.html
- How to contact us?
  - {Daniel.Slamanig, Christoph.Striecks}@ait.ac.at
- Tutor: Karen Klein
  - karen.klein@ist.ac.at

- Official page at TU, Location etc.
  - https://tiss.tuwien.ac.at/course/courseDetails.xhtml?dswid=8632&dsrid=679&courseNr=192062&semester=2018W
- Tutorial, TU site
  - https://tiss.tuwien.ac.at/course/courseAnnouncement.xhtml?dswid=5209&dsrid=341&courseNumber=192063&courseSemester=2018W
- Exam for the second part: Thursday 31.01.2019 15:00-17:00 (Tutorial slot)

Crypto 2.0: Public-Key Encryption



Properties:

- Enables secure one-to-one communication
- Solves key-distribution problem (pk is publicly available) compared to secret-key encryption
- Key pk has to be authenticated (e.g., by using heavy Public-Key Infrastructures)
- Encryption is all-or-nothing

Recall Public-Key Encryption

- Gen(1^n): on input security parameter 1^n, return public and secret keys (pk, sk), where message space *M* is defined in pk.

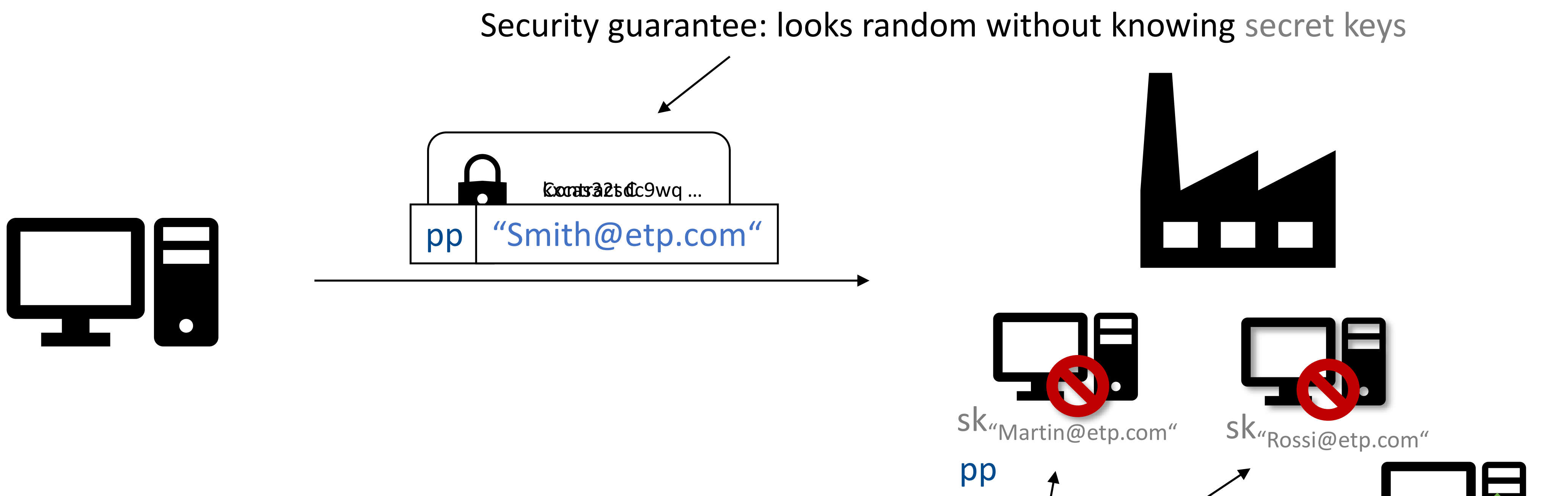
- Enc(pk,m): on input public key pk and message m, return ciphertext c

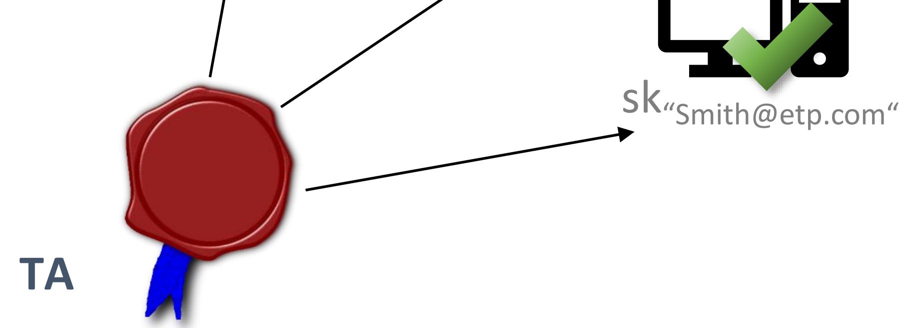
- Dec(sk,c): on input secret key sk and ciphertext c, return m or error

- Correctness: for all integer k, for all (pk,sk) $\leftarrow$ Gen(1^n), for all messages m, for all c $\leftarrow$ Enc(pk,m), we have that m = Dec(sk,c) holds except with negl. probability.

- Security: OW-CPA, IND-CPA, IND-CCA notions

Crypto 3.0: Identity-Based Encryption





Properties:

- Public key is a public string, e.g., email address
  - Essentially compress exponentially many pk’s into short pp
- Many user secret keys for one pp („less-heavy“ PKI needed)
- Need of pp-related authority **TA** that distributes keys

Identity-Based Encryption, Definition*

DEFINITION. A IBE scheme Ξ with **identity** and message spaces ***ID*** and *M*, respectively, consist of four PPT algorithms (Gen, **Ext**, Enc, Dec) such that:

- Gen(1^k): on input security parameter 1^k, return public parameters and secret key (pp, sk), where message space *M* **and identity space *ID*** is defined in pp.
- **Ext(sk,id): on input identity id and secret key sk, return user secret key $\mathbf{usk_{id}}$.**
- Enc(pp,m,**id**): on input public parameter pp, **identity id ϵ *ID***, and message m ϵ *M*, return ciphertext $\mathbf{c_{id}}$
- Dec($\mathbf{usk_{id}}$,$\mathbf{c_{id}}$): on input secret key $\mathbf{usk_{id}}$ and ciphertext $\mathbf{c_{id}}$, return m or error
- **Correctness**: for all integer k, for all (pp,sk)←Gen(1^k), **for all identities id ϵ *ID***, **for all $\mathbf{usk_{id}}$**← **Ext(sk,id)**, for all messages m **ϵ** *M*, for all $\mathbf{c_{id}}$←Enc(pp**,id**,m), we have that m=Dec($\mathbf{usk_{id}}$,$\mathbf{c_{id}}$) holds except with negl. probability.
- **Security**: IBE-IND-CPA and IBE-IND-CCA notions (plus variants thereof)

*We highlight the main differences to PKE with **bold**.

Some Remarks on the IBE Definition

- As in PKE, encryption may be deterministic or probabilistic
- As in PKE, decryption may be perfectly correct or may fail with negl. probability

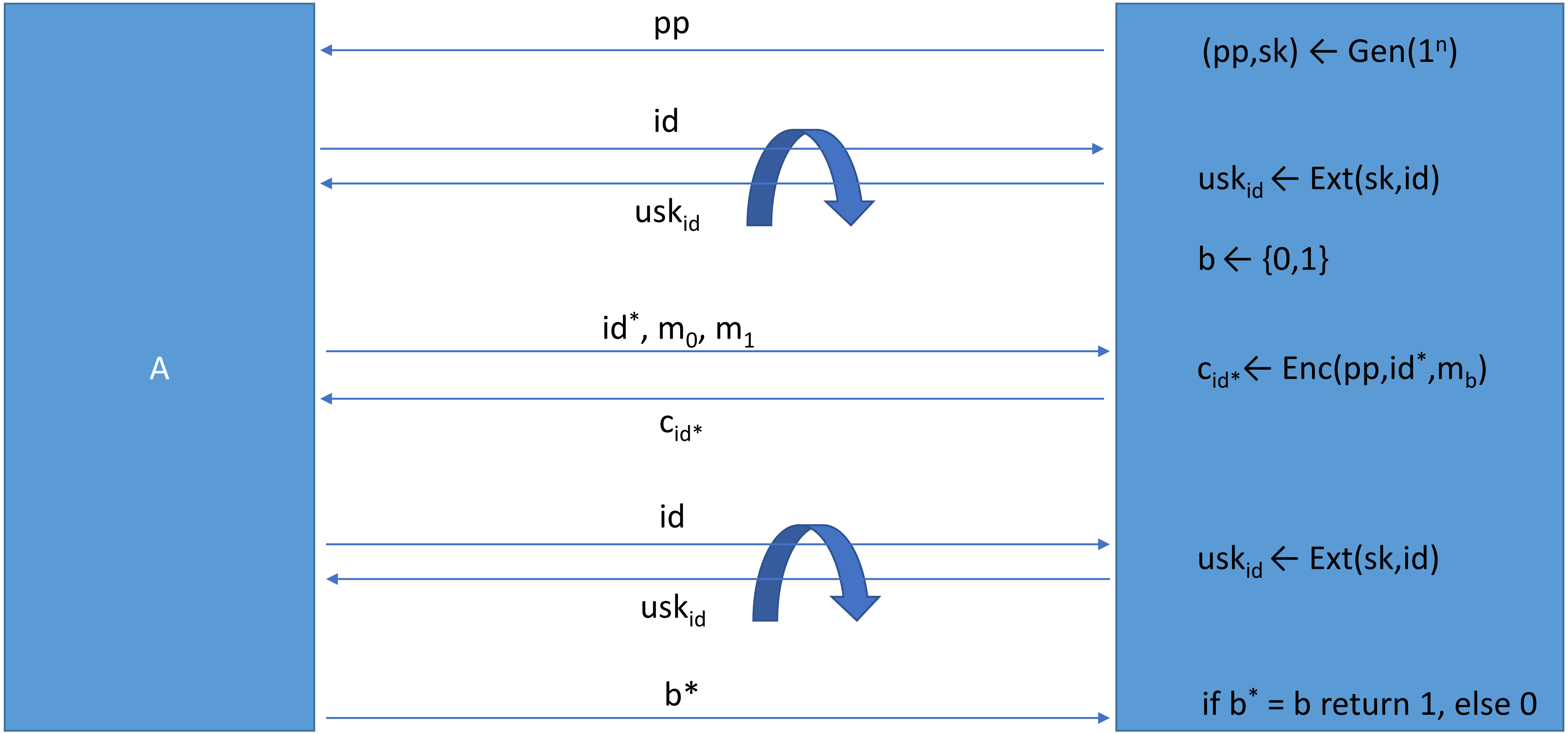
- Opposed to PKE, an identity space is defined which is typically exponentially large (question: why?)
  - This also means exponentially many user secret keys possible and, hence, constitute a multi-user encryption system

  - But: trusted authority is needed to generate user secret keys

Security Definitions (Initial Thoughts)

- IBE scheme is a multi-user system
  - Multiple user secret keys can be compromised
  - Attacker should be able to retrieve user secret keys of its choice (not the case in IND-CPA security)
  - Similarly to IND-CPA, attacker should not be able to distinguish ciphertexts of chosen messages and “target identity” (question: what must be realized by a security definition to exclude trivial wins?)
  - We will dub the security notions for IBE as IBE-IND-CPA

IBE-IND-CPA Security: $\mathrm{Exp}_{\mathrm{IBE},A}$

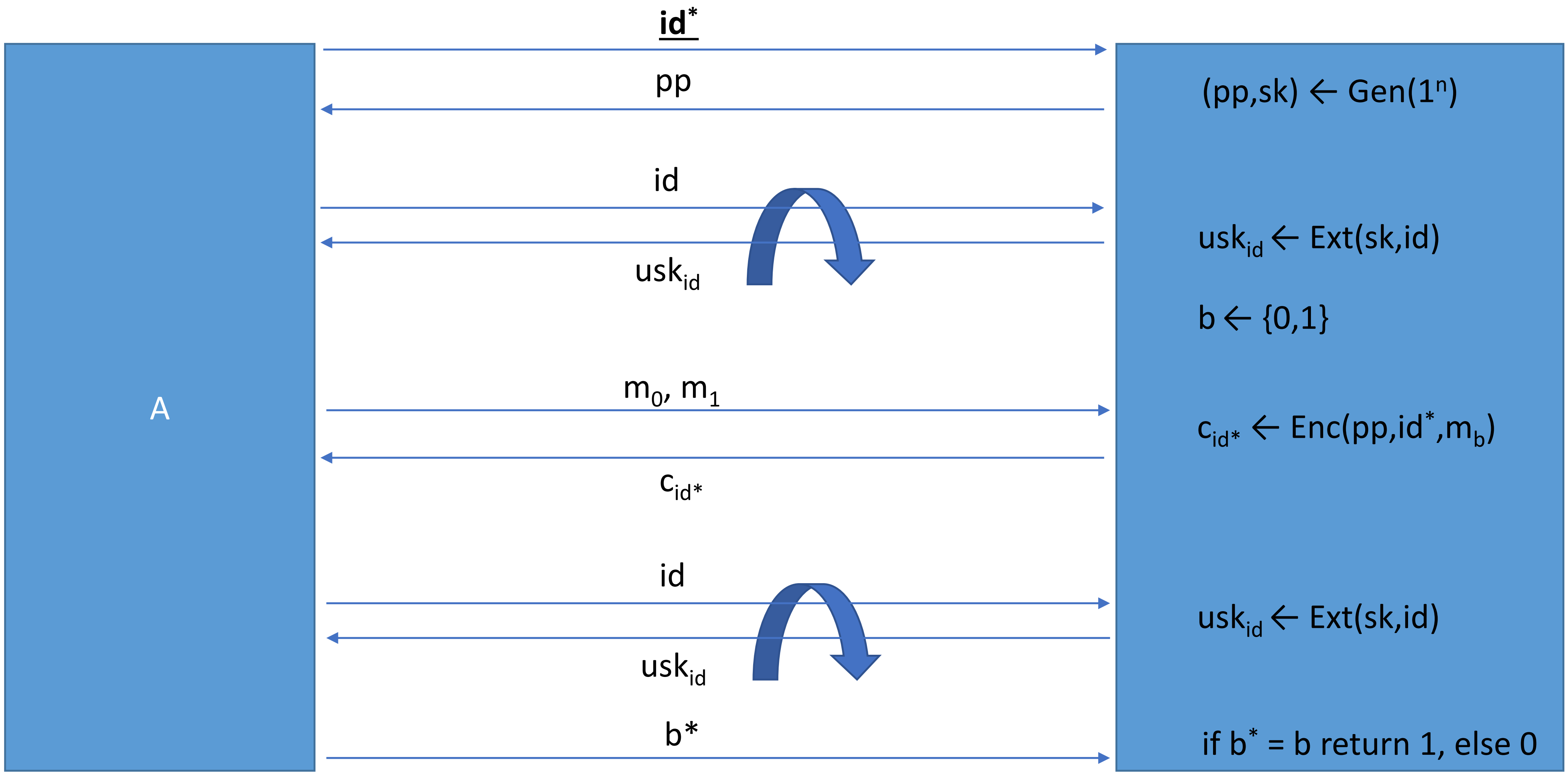


IBE-IND-CPA Security

Definition. An IBE scheme Ξ = (Gen, Ext, Enc, Dec) is IBE-IND-CPA secure if and only if $\mathrm{Adv}_{IBE,A}(1^n) := |\Pr[\mathrm{Exp}_{IBE,A}(1^n)=1] - ½|$ is negl. in n, for any valid PPT adversary *A* and $|m_0|=|m_1|$. *A* is valid if id^* was never queried by *A*.

- Remark: IBE-IND-CPA security is very hard to achieve
- That is the reason why the first schemes were only proven secure in a weaker model dubbed Weak-IBE-IND-CPA

Weak-IBE-IND-CPA Security: $\mathrm{Exp}_{\text{Weak-IBE},A}$



Weak-IBE-IND-CPA Security

Definition. An IBE scheme Ξ = (Gen, Ext, Enc, Dec) is Weak-IBE-IND-CPA secure if and only if $Adv_{Weak\text{-}IBE,A}(1^n) := |Pr[Exp_{Weak\text{-}IBE,A}(1^n)=1] - ½|$ is negl. in n, for any valid PPT adversary *A* and $|m_0|=|m_1|$. *A* is valid if id^* was never queried by *A*.

- Indeed, many system in the literature were constructed to be "only" Weak-IBE-IND-CPA secure
  - IBE-IND-CPA in Standard Model (without RO) hard to achieve (only 2005 with large parameters)
  - However, *inefficient* generic transformations (from Weak-IBE-IND-CPA to IBE-IND-CPA) are known

Constructing IBE

- Interestingly, constructing IBE is harder compared to constructing PKE
  - Mathematical “trick” necessary, i.e., **pairing**
  - Up to now, only a few (inefficient) schemes exist that do not rely on pairings (e.g., best paper from CRYPTO 2017 under DDH, or Cocks’ scheme from factoring)

- Proposed by Shamir in the 1984, first realizations only 2001 due to Boneh and Franklin and Cocks

- IBE is building block for: digital signatures, searchable encryption, IND-CCA secure PKE, forward-secret encryption

Strong Tool: Pairings

- Given cyclic groups G, G_T with prime-order p
- Furthermore, given a mapping e: G x G -> G_T and generator g ϵ G

- Properties
  - Non-degeneracy: for all g ϵ G, g ≠ 1, e(g,g) ≠ 1 holds.
  - Bilinearity: for all g ϵ G and integers a,b, $e(g^a,g^b) = e(g^b,g^a)$ holds. (In part., e(g,h)=e(h,g), why?)
- DDH assumption might not hold in G, since one can efficiently test DDH tuples (as a result, Bilinear DH assumption was introduced, also used in IBE constructions and further)

Boneh-Franklin (BF) IBE

- Assume (e, G, G_T, p, g) and Random Oracle $H : ID \to G$, message and identity spaces *M* and *ID*, resp., are given as input to each algorithm
- $BF.Gen(1^k)$: return $(pp, sk) := (g^x, x)$, for g in G
- $BF.Ext(id, sk)$: return $usk_{id} := H(id)^x$
- $BF.Enc(pp, id, m)$: return $c_{id} := (c_1, c_2) := (g^y, e(g^x, H(id))^y * m)$
- $BF.Dec(usk_{id}, c_{id})$: return $c_2 / e(c_1, usk_{id})$
- Correctness holds (why?):
  - $e(c_1, usk_{id}) = e(g, H(id))^{xy}$ and $e(g, H(id))^{xy} * m = c_2$

Bilinear Diffie-Hellman Assumption

- Bilinear DH (BDH) assumption is an extension of the computational DH assumption to the pairing setting
  - Essentially: given g^x, g^y, g^z it is hard to compute $e(g,g)^{xyz}$

- Security of BF IBE: IBE-IND-CPA secure in the RO model under BDH assumption

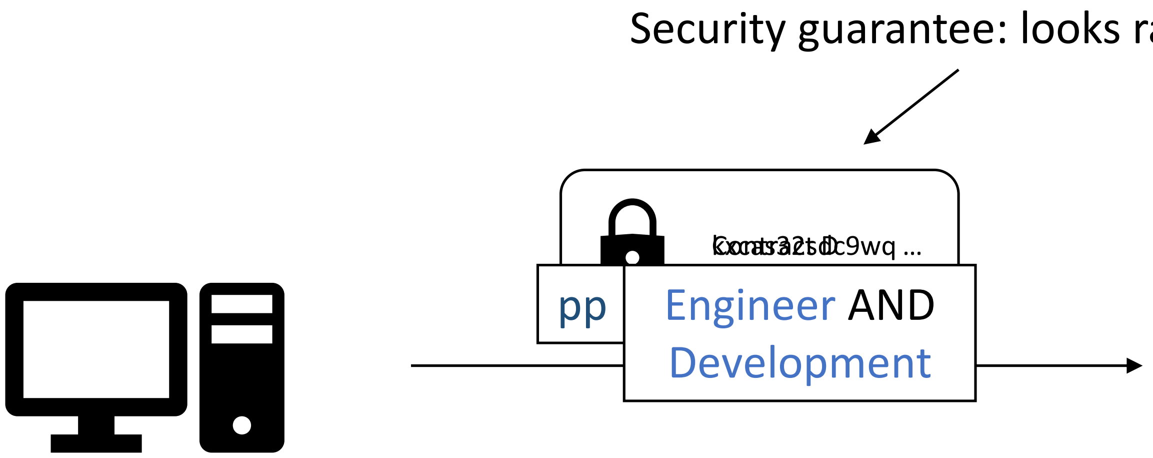
- Many schemes in the Standard-Model were only proven Weak-IBE-IND-CPA secure until 2009 (Waters)
- Nowadays: many IBE-IND-CPA and IBE-IND-CCA schemes are known and constitute state-of-the-art

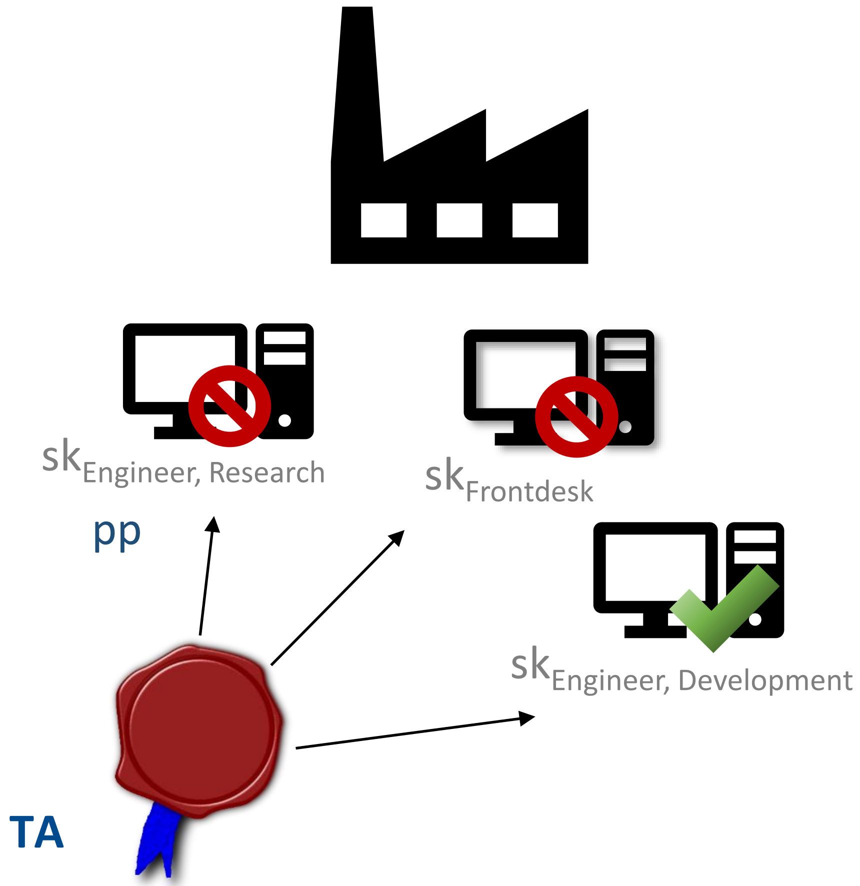
Naor’s Tranformation

- Interesting observation: each IBE scheme is also a signature scheme due to Naor (described in Boneh-Franlin IBE paper from 2001)

- Sketch:
  - Signature public and secret keys (pp, sk) are public parameter and secret key of the IBE.Gen, respectively
  - The signature σ is the output of IBE.Ext with “identity” m and sk (where m is the message in the signature scheme)
  - Verification of a signature σ and a message m is done by running IBE.Enc with pp and random message R and m; and try decrypting the resulting ciphertext with the signature σ
  - If the result of the decryption yields R, then the signature is valid for m under pp
  - Correctness? Homework …

Crypto 4.0: Attribute-Based Encryption





Properties:

- Enables **fine-grained** one-to-many communication
- Enforces access control on the cryptographic level, collusion-safe
- Many secret keys for one pk (hence, less-heavy PKI needed)
- Need of pp-related authority **TA** that distributes secret keys

Attribute Based Access Control

https://csrc.nist.gov/projects/abac/

NIST

Information Technology Laboratory

COMPUTER SECURITY RESOURCE CENTER

PROJECTS

Attribute Based Access Control

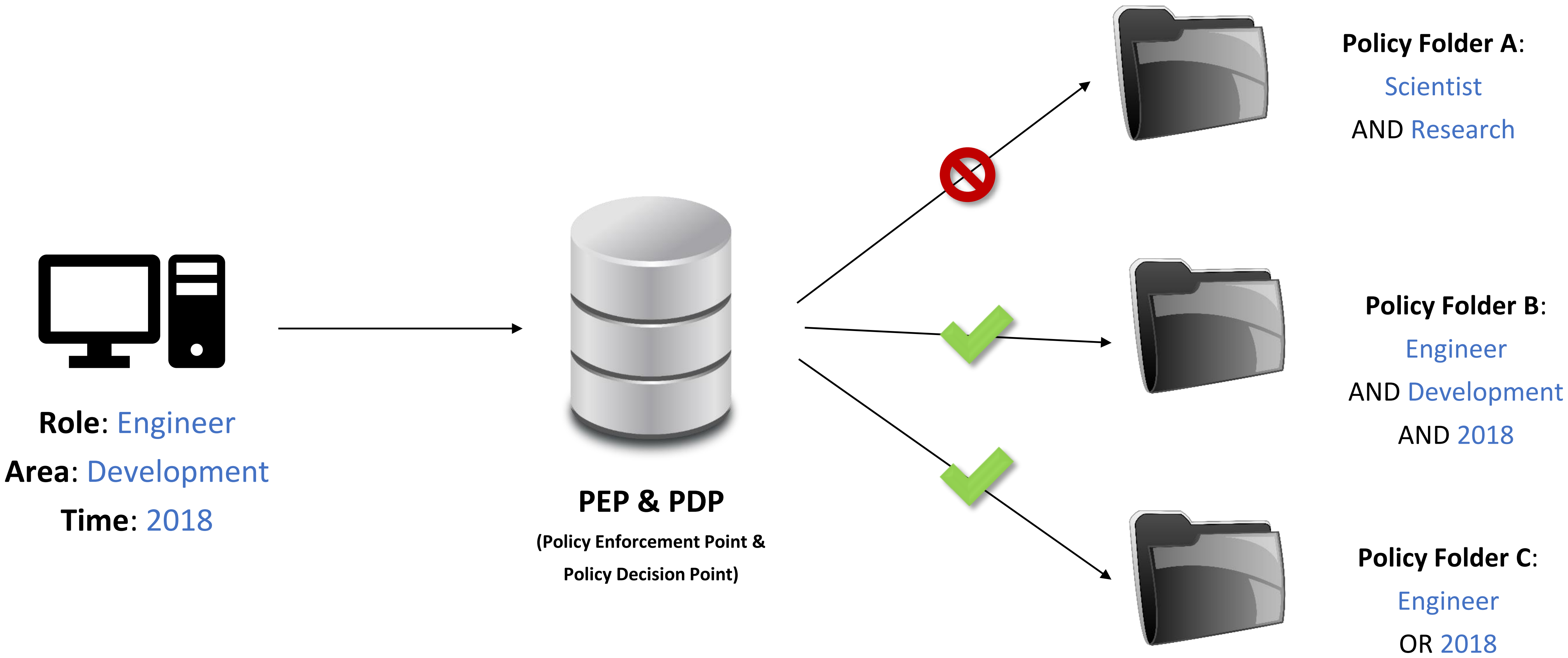
Project Overview

The cor ... um of
logical ... ible
methoc

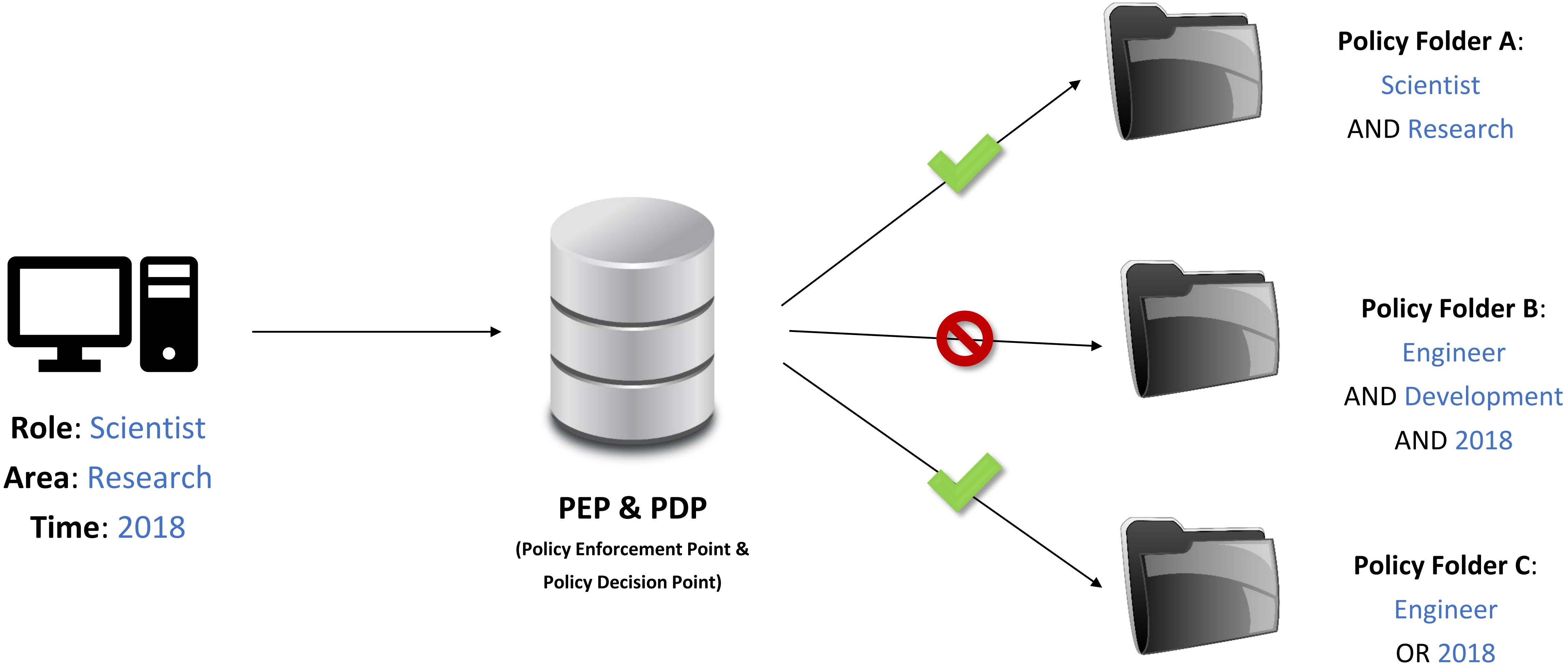
In Nove ... ity,
Creden ... federal
organiz ... enable
access within and between organizations across the Federal enterprise. In December 2011, the FICAM Roadmap and Implementation Plan v2.0 took the next step of calling out ABAC as a recommended access control model for promoting information sharing between diverse and disparate organizations.

NIST Special Publication 800-162 (Jan. 2014)

Attribute-Based Access Control (ABAC, simplified)



Attribute-Based Access Control (ABAC, simplified)



Attribute-Based Access Control (ABAC)

- Advantage: fine-grained access to data, defined on attributes and policies with strong PEP/PDP mechanisms
- Disadvantage: massive trust in software-based PEP/PDP implementations (software implementation often prone to errors)

Can we do better?

Yes! Enforcing access control through cryptography using Attribute-Based Encryption (ABE)

Initial Thoughts on ABE

- Attributes and Policies play essential part in ABE
  - Attributes are (bit) strings
  - Policies can be seen as Boolean formulas, e.g., („Scientist“ AND „Research“) OR „Engineer“
  - Informal for now: we say „an attribute set satifies a policy“ if the Boolean formula evaluates to true for an attribute input set

- Where to put attributes? Ciphertext, Keys?
- Where to put policies? Ciphertext, Keys?

- Hence, two variants of ABE exist
  - **Key-Policy ABE (KP-ABE):** ciphertexts hold attributes, keys hold policies
  - **Ciphertext-Policy ABE (CP-ABE):** ciphertexts hold policies, keys hold attributes

KP-Attribute-Based Encryption, Definition

DEFINITION. A KP-ABE scheme Ω_{KP} consist of four PPT algorithms (Gen, **Ext**, Enc, Dec) such that:

- Gen(1^k): on input security parameter 1^k, return public parameters and secret key (pp, sk), where message space *M* **and attribute space *A*** and **policy space *P*** is defined in pp.
- **Ext(sk,p): on input secret key and policy p ϵ *P*, return user secret key usk_p.**
- Enc(pp,**a**,m): on input public parameter pp, **attribute set a ϵ *A***, and message m ϵ *M*, return ciphertext c_a
- Dec(usk_p,c_a): on input secret key usk_p and ciphertext c_a, return m **if a satisfies p**, or error
- Correctness: for all integer k, for all (pp,sk) ← Gen(1^k), **for all attribute sets a ⊂ *A***, **for all policies p ϵ *P***, **for all usk_p** ← **Ext(sk,p)**, for all messages m, for all c_a ← Enc(**a**,**pp**,m), we have that m=Dec(usk_p ,c_a) holds **if a satisfies p** except with negl. probability.
- Security: KP-ABE-IND-CPA (on slide 28), KP-ABE-IND-CCA notions

CP-Attribute-Based Encryption, Definition

DEFINITION. A CP-ABE scheme Ω_{KP} consist of four PPT algorithms (Gen, **Ext**, Enc, Dec) such that:

- Gen(1^k): on input security parameter 1^k, return public parameters and secret key (pp, sk), where message space *M* **and attribute space *A*** and **policy space *P*** is defined in pp.
- **Ext(sk,a): on input secret key and attribute set a ϵ *A*, return user secret key usk_a.**
- Enc(pp,**p**,m): on input public parameter pp, **policy p ϵ *P***, and message m ϵ *M*, return ciphertext $\mathbf{c_p}$
- Dec($\mathbf{usk_a}$,$\mathbf{c_p}$): on input secret key $\mathbf{usk_a}$ and ciphertext $\mathbf{c_p}$, return m **if a satisfies p**, or error
- Correctness: for all integer k, for all (pp,sk) ← Gen(1^k), **for all attribute sets a $\subset$ *A***, **for all policies p ϵ *P***, **for all usk_a** ← **Ext(sk,a)**, for all messages m, for all $\mathbf{c_p}$ ← Enc(pp,**p**,m), we have that m = Dec($\mathbf{usk_a}$,$\mathbf{c_p}$) holds **if a satisfies p** except with negl. probability.
- Security: CP-ABE-IND-CPA (on slide 30), CP-ABE-IND-CCA notions

Some Remarks on the Definition

- As in IBE, encryption may be deterministic or probabilistic
- As in IBE, decryption may be perfectly correct or may fail with negl. probability
- As in IBE, exponentially many user secret keys possible and, hence, constitute a multi-user encryption system

- Opposed to IBE, an attributes space **and** a policy space is defined
- As in IBE, trusted authority is needed to generate user secret keys

Security Definitions (Initial Thoughts)

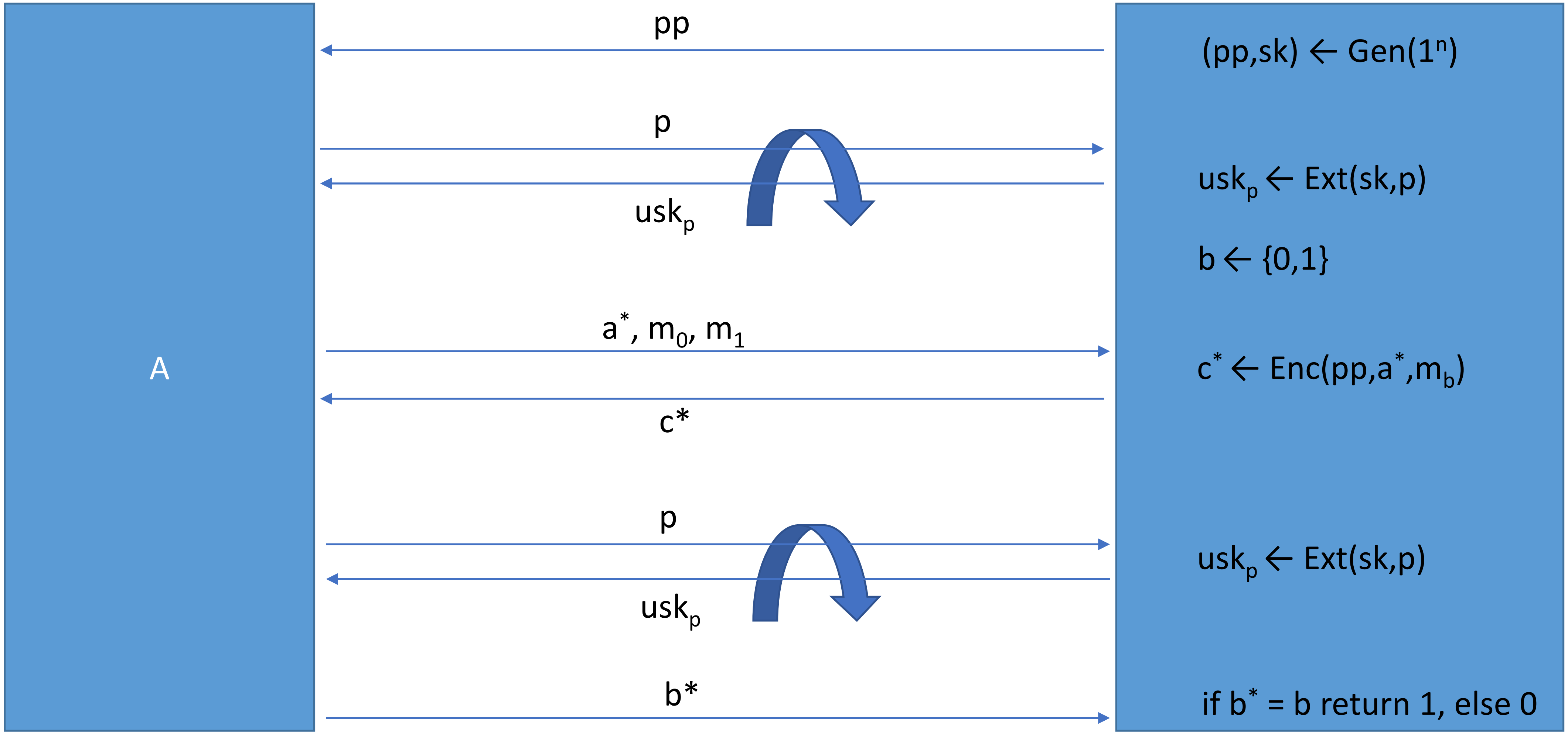
- IBE scheme is a multi-user system
  - Multiple user secret keys can be compromised (and combined)
  - Distinguishing feature in ABE: collusion resistance!

  - Attacker should be able to retrieve user secret keys of its choice

  - Similarly to IBE-IND-CPA, attacker should not be able to distinguish ciphertexts of chosen messages and “attribute set” of “policy” (question: what must be realized by a security definition to exclude trivial wins?)

  - We will dub the security notions for KP-ABE and CP-ABE as KP-ABE-IND-CPA and CP-ABE-IND-CPA, respectively

KP-ABE-IND-CPA Security: $Exp_{KP\text{-}ABE,A}$

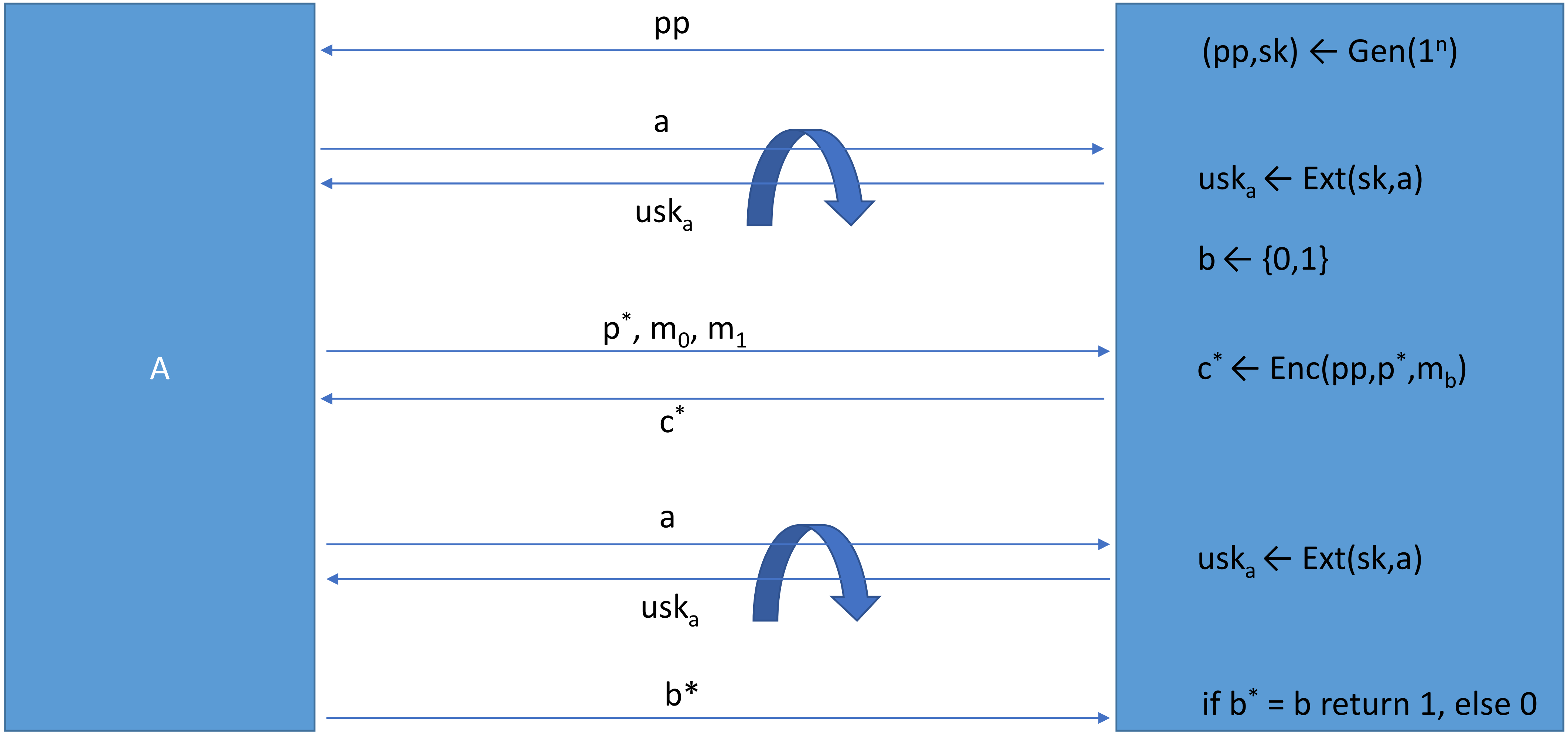


KP-ABE-IND-CPA Security

Definition. An KP-ABE scheme Ω = (Gen, Ext, Enc, Dec) is KP-ABE-IND-CPA secure if and only if $Adv_{KP\text{-}ABE,A}(1^n) := |Pr[Exp_{KP\text{-}ABE,A}(1^n)=1] - ½|$ is negl. in n, for any valid PPT adversary A and $|m_0|=|m_1|$. A is valid if a^* does not satisfy any queried policy.

- Remark: KP-ABE-IND-CPA security is very hard to achieve indeed
- Similar to IBE, the first ABE schemes were only proven secure in a weaker model

CP-ABE-IND-CPA Security: $Exp_{CP\text{-}ABE,A}$



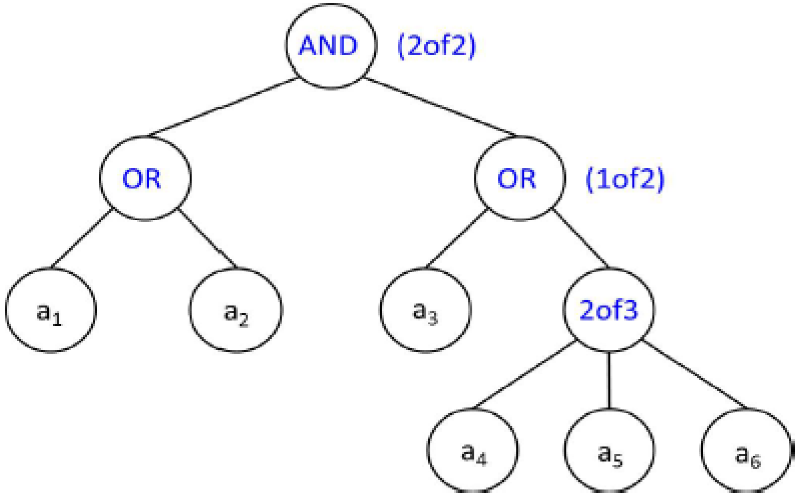
CP-ABE-IND-CPA Security

Definition. An CP-ABE scheme Ω = (Gen, Ext, Enc, Dec) is CP-ABE-IND-CPA secure if and only if $\mathrm{Adv}_{\text{CP-ABE},A}(1^n) := |\Pr[\mathrm{Exp}_{\text{KP-ABE},A}(1^n)=1] - ½|$ is negl. in n, for any valid PPT adversary A and $|m_0|=|m_1|$. A is valid if any queried a does not satisfy p^*.

- Remark: CP-ABE-IND-CPA security is very hard to achieve as well

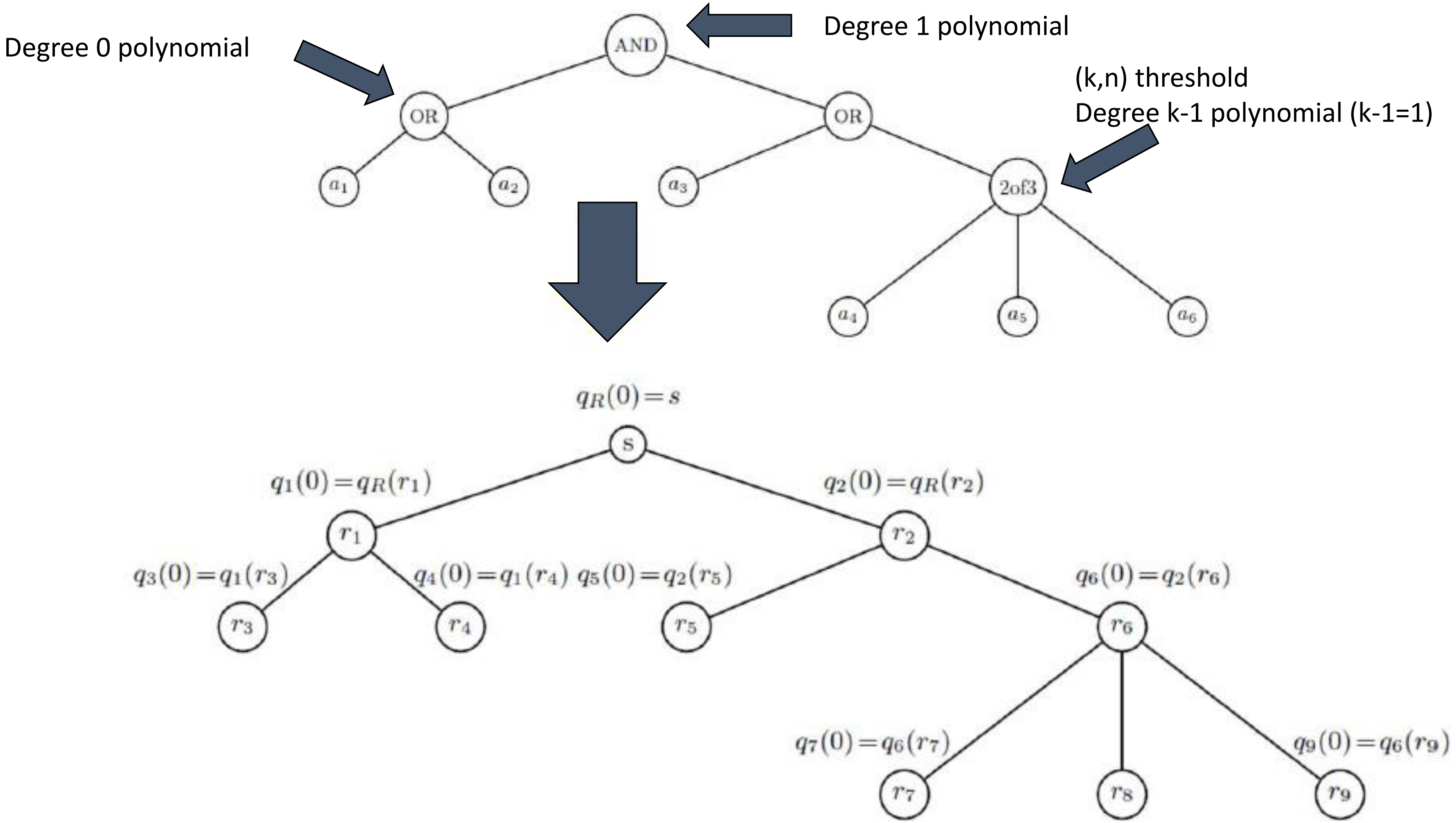
Constructing CP-ABE (Bethencourt, Sahai, Waters, IEEE S&P 2007)

- The policy is put into the ciphertext and user secret keys are issued for sets of attributes
- Main techniques: „access trees“, pairings, and secret sharing
  - Let A = $\{a_1,\ldots,a_6\}$ be the set of attributes, with policy p=(a_1 **OR** a_2) **AND** (a_3 **OR** **2of3**(a_4,a_5,a_6))*:



* Here, we also allow a threshold gate **2of3**.

CP-ABE Idea (BSW07)



CP-ABE Idea (BSW07)

- Public key (system parameters)

$$pk = (g, h = g^{\beta}, e(g,g)^{\alpha})$$

- User with attribute set A = {a_1,…,a_n} gets private key

$$(D = g^{(\alpha+r)/\beta}, (D_i = g^r \cdot H(a_i)^{r_i}, D_i' = g^{r_i})_{a_i \in A})$$

- Keys are randomized per user $(r, r_1, \ldots, r_n)$ to avoid collusion attacks
- Ciphertexts for policy (access tree) including all leafs j and with root of tree

$$q_R(0) = s$$

$$C' = m \cdot e(g,g)^{\alpha s}, C = h^s, (C_j = g^{q_j(0)}, C_j' = H(a_j)^{q_j(0)})$$

CP-ABE Idea (BSW07)

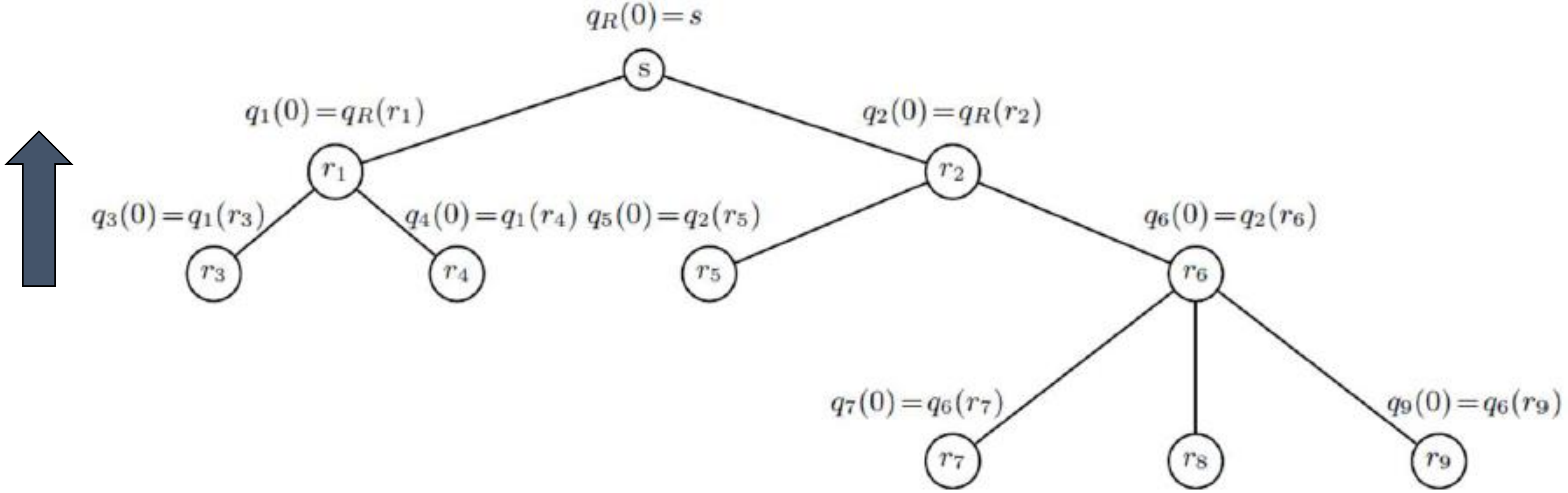
- Decryption: Start at the leaves

$$\begin{aligned}
\mathsf{DecryptNode}(c, sk, j) &= \frac{e(D_j, C_j)}{e(D'_j, C'_j)} \\
&= \frac{e(g^r \cdot H(j)^{r_j}, g^{q_j(0)})}{e(g^{r_j}, H(j)^{q_j(0)})} = \frac{e(g,g)^{rq_j(0)} e(g^{r_j}, H(j)^{q_j(0)})}{e(g^{r_j}, H(j)^{q_j(0)})} \\
&= e(g,g)^{rq_j(0)}.
\end{aligned}$$

- Work up the tree for all inner nodes, then remove masking
- Polynomial interpolation in the exponent
- Works if user secret key contains attributes such that the threshold of every inner node can be satisfied

CP-ABE Idea (BSW07)

$$A = e(g,g)^{r \cdot q_R(0)} = e(g,g)^{r \cdot s}$$



$$C'/(e(C,D)/A) = C'/(e(h^s, g^{(\alpha+r)/\beta})/e(g,g)^{rs}) = C'/e(g,g)^{\alpha s} = m$$