“Security” Concept for Peer-to-Peer Systems

Stefan.Kraxberger@iaik.tugraz.at
Udo.Payer@iaik.tugraz.at

http://www.secricom.eu

SECRICOM
Seamless Communication for Crisis Management

… is a research project creating Seamless Communication for Crisis Management

Objectives:

- Seamless and secure interoperability of mobile devices
- Creation of pervasive and trusted communication infrastructure
- Provide true collaboration and inter-working of emergency responders

"Security" Concept for Peer-to-Peer Systems

Stefan.Kraxberger@iaik.tugraz.at
Udo.Payer@iaik.tugraz.at

Heterogeneous P2P System

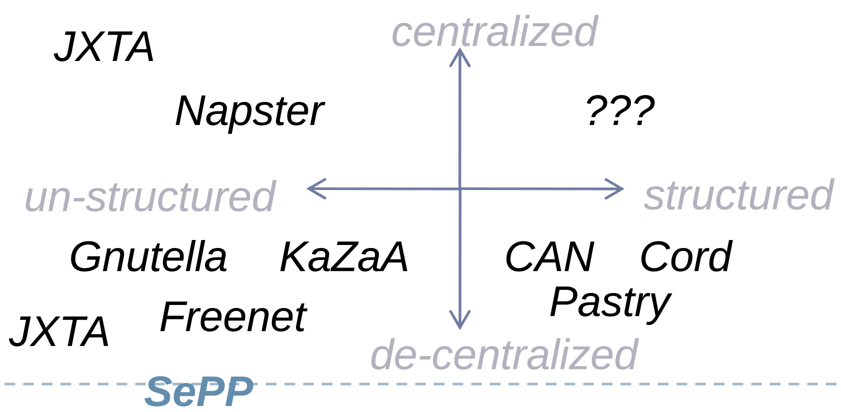
… **focused on topics like:**

- Distributed storage
- Distributed computing
- Content delivery/streaming
- Messaging/Telephony (PTT)

Heterogeneous P2P System

Unstructured-Decentralized:

- **Centralized** P2P network such as Napster
- **Decentralized** P2P network such as KaZaA
- **Structured** P2P network such as CAN (any node can efficiently route a search to some peer)
- **Unstructured** P2P network such as Gnutella (2-tier overlay links are established arbitrarily)
- **Hybrid** P2P network (Centralized and Decentralized) such as JXTA



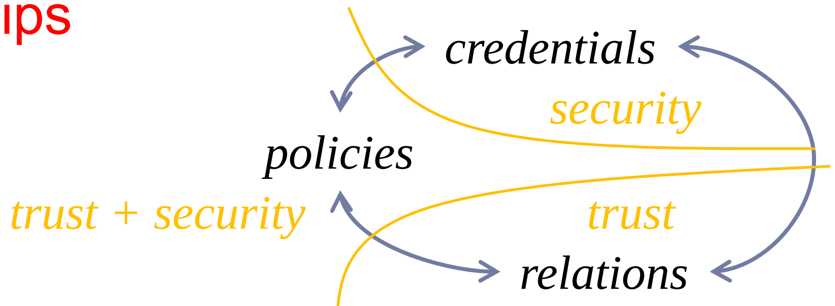
http://sourceforge.net/projects/secureP2P

Attacks to P2P Systems

- What attacks:
  - Poisoning distributed index
  - Poisoning overlay routing tables
  - Identity attacks (Sybil attack)
    - book by Flora Rheta Schreiber (1973) about the treatment of Sybil Dorsett
  - *Byzantine* Generals Problem
  - Eavesdropping (Wormhole attack)
  - Pollution attack (*polluted content can spread through much of the P2P network*)
  - Denial of service
  - ...
- Why?
  - No centralized node acts as an *authority*
  - Absence of a defensible border: *friend or foe*
    - At network level: *break routing system*
    - At application level: *Corrupt or delete data can be forwarded*

Trust Models

- "Trust Management" *was first coined by* Blaze et. al 1996 *)
  - coherent framework for the study of security policies, security credentials and trust relationships



  - The first TM systems: PolicyMaker and KeyNote.
- 3 "Trust Management" models
  - Certificate-based
  - Policy-based
  - Reputation-based (*behaviour observed directly or indirectly*)
    - *trust information is shared among peers*

*) Decentralized Trust Management

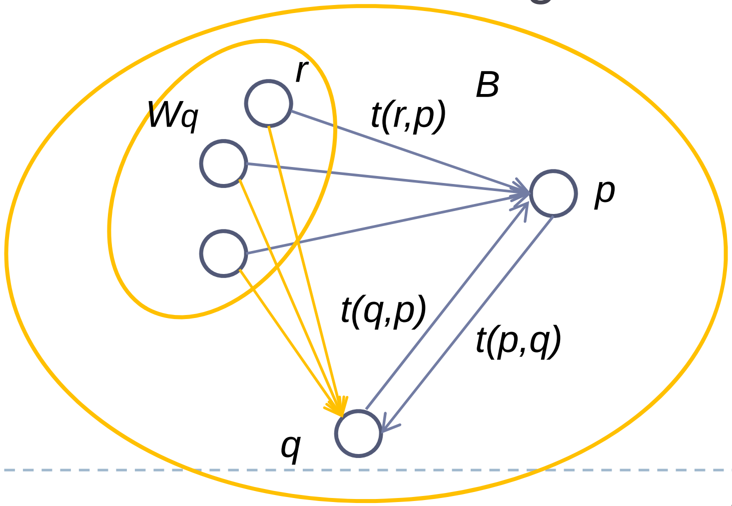
P2P Trust-Models

- Global Trust Model (*Aberer, K. and Despotovic, Z. , 2001*)
  - Based on reputation in a distributed system
  - Reputation := statistical data mining process to find out if an agent will cheat (*analysis of former transactions*).

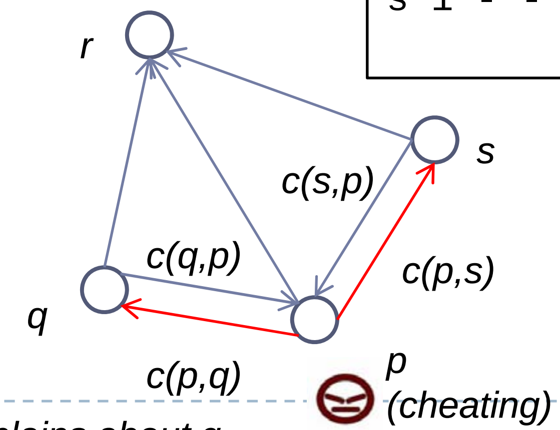
all agents

global behaviour: $B(p) = \{t(p,q) \;\text{ or }\; t(q,p) \mid q \in P\} \subseteq B$

  - Trust Model is worthless, if it is based on a centralized database
  - local direct knowledge $B_q(p) = \{t(q,p) \mid t(q,p) \in B\}$
  - local indirect knowledge $W_q(p) = \{t(r,p) \mid r \in W_q, t(r,p) \in B\}$



But:



	p	q	s	
p	-	1	1	Σ=2
q	1	-	-	Σ=1
s	1	-	-	Σ=1

c(p,q) ... p complains about q

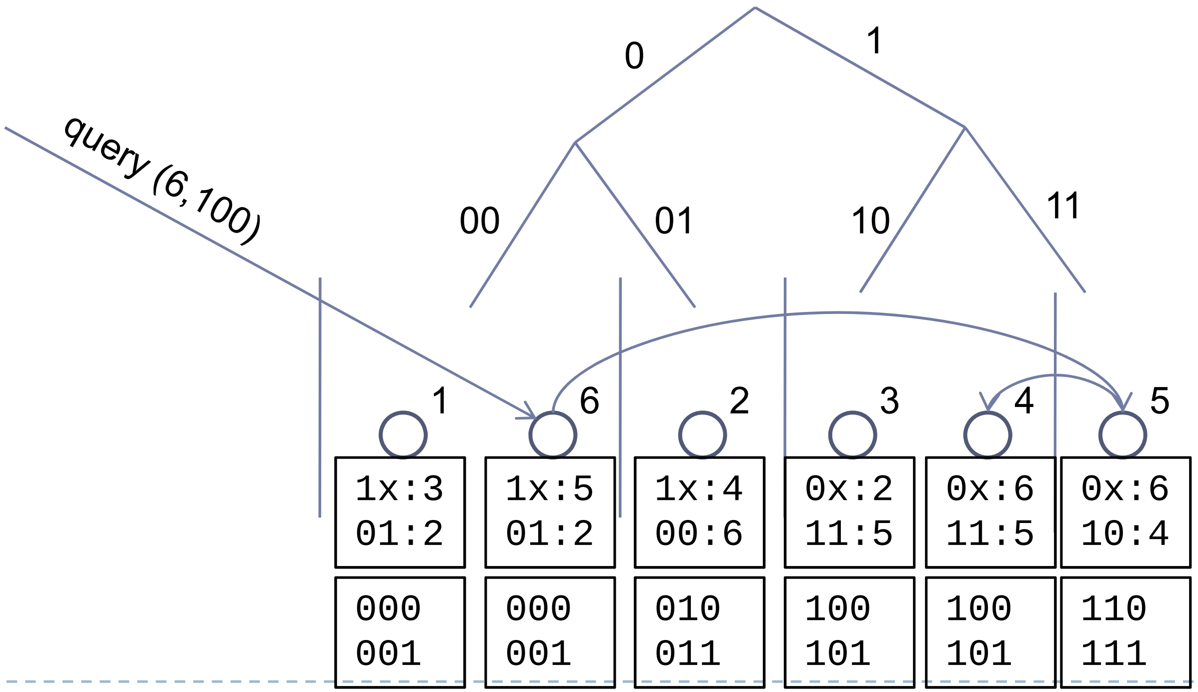
P2P Trust-Models

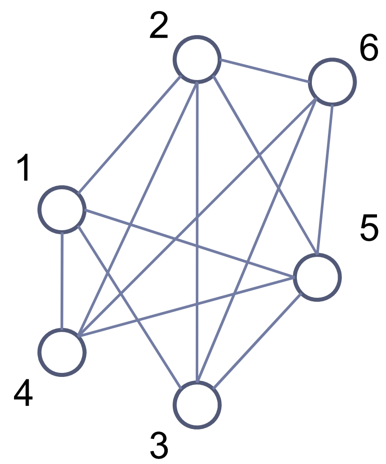
$$\begin{bmatrix}1\\1\\0\\1\\1\end{bmatrix} \qquad 4\ \begin{bmatrix}1 & 1 & 0 & 1 & 1\end{bmatrix}$$

- Global Trust Model *(Aberer, K. and Despotovic, Z. , 2001)*
  - reputation (node) := $T(p) = |\{c(p,q) \mid q \in P\}| \times |\{c(q,p) \mid q \in P\}|$ *trustworthiness*
  - Decentralised management
    - … in this case based on P-Grid

$$\begin{bmatrix}0\\0\\0\\1\\0\end{bmatrix} \qquad 1\ \begin{bmatrix}0 & 0 & 0 & 1 & 0\end{bmatrix}$$

(*DHTs work efficiently only for uniform load-distributions*)





P2P Trust-Models

- Global Trust Model (*Aberer, K. and Despotovic, Z. , 2001*)
  - minimal global agreement: *key space*
  - search can be done in *O(log(n))*
  - *local computation of trust:*
    - *send query(a, key(q)) s times … p optains:* $W = \{(cr_i(q), cf_i(q), a_i, f_i) \mid i = 1, \ldots, w\}$

#complaint _q_rec(a_i)

#complained_q_filed_q (a_i)

frequency(a_i)

# of witnesses

prob. of not finding witness i

$$cr_i^{norm}(q) = cr_i(q)(1 - (\frac{s - f_i}{s})^s), \quad i = 1, \ldots, w$$

- *compute level of trust (1= trustworthy, -1= mistrust)*

$$decide_p(cr_i^{norm}(q), cf_i^{norm}(q)) =$$

if

$$cr_i^{norm}(q) cf_i^{norm}(q) \leq (\frac{1}{2} + \frac{4}{\sqrt{cr_p^{avg} cf_p^{avg}}})^2 cr_p^{avg} cf_p^{avg}$$

then 1 **else** -1

P2P Trust-Models

- EigenTrust (*Sep Kamvar, et al., 2003*)
  - based on the notion of *transitive trust* :
    - a peer will have a high opinion of those peers who have provided *authentic files*
  - Designed for reputation management of P2P Systems
  - The global reputation of each peer is marked by the local trust values assigned by other peers

$$s_{ij} = \overbrace{\mathrm{sat}(i,j)}^{\#satisfactory} - \overbrace{\mathrm{unsat}(i,j)}^{\#unsatisfactory}$$

$$i \xrightarrow{s_{ij}} j$$

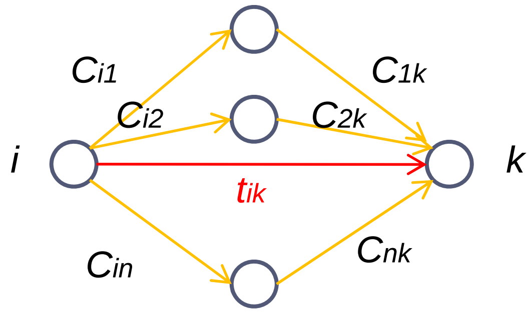
$$\begin{bmatrix} & & s\_1j & & \\ & & s\_2j & & \\ & & : & & \\ s\_i1 & s\_i2.. & s\_ij & .. & s\_in \\ & & : & & \\ & & s\_nj & & \end{bmatrix}$$

  - Normalize c_{ij} since malicious peers can cheat

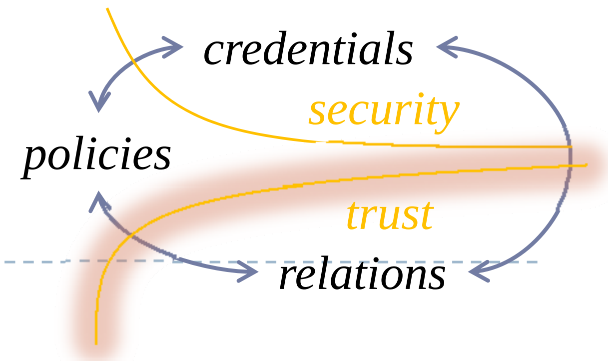
$$c_{ij} = \frac{\max(s_{ij}, 0)}{\sum_j \max(s_{ij}, 0)}$$

P2P Trust-Models

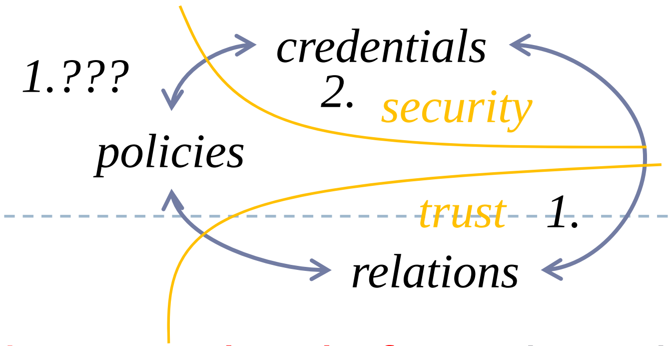
- EigenTrust (*Sep Kamvar, et al., 2003*)
  1. Local trust value: $t_{ik} = \sum_j c_{ij} c_{jk}$
  2. … based on $\sum_j t_{ij} = 1$
  3. … we can write: $t = C^T . c_i$
  4. … peer may wish to ask his friend's friend: $(t = (C^T)^2 c_i)$
  5. … and so on until all nodes are contacted: $(t = (C^T)^n c_i)$
  6. t will converge to the same vector for all peers i !!!
  7. … the left principal eigenvector of C
  8. … due to 6. we can compute $\vec{t} = (C^T)^n \vec{e}$ for large n
  9. and get:



$\vec{t}^{(0)} = \vec{e};$
repeat
$\quad \vec{t}^{(k+1)} = C^T \vec{t}^{(k)};$
$\quad \delta = ||t^{(k+1)} - t^k||;$
until $\delta < \epsilon;$



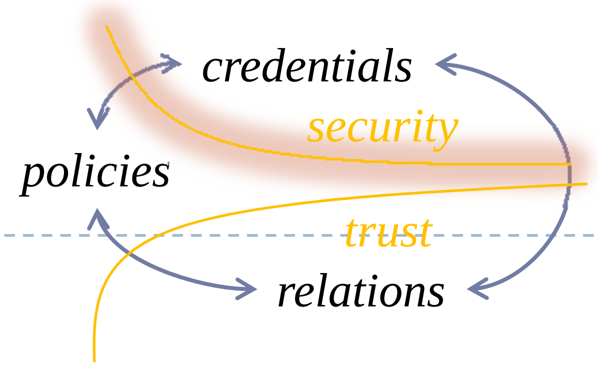
Trust vs. Security



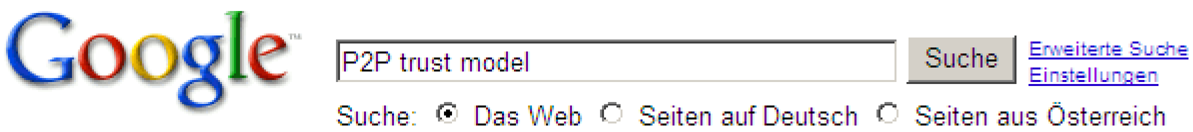
- trust preexists security *)
  - all security mechanisms require some level of trust in various components of the system
  - *security mechanisms* can help to *transfer* trust in one component to trust in another component, but they *cannot create trust by themselves*
- cooperation reinforces trust *)
  - trust is about the ability to *predict* the behaviour of another party
  - ... in the broader sense: BIOS + (CRTM + TPM) → ...
    - (*i.e., follow certain rules for the benefit of the entire system*) makes predictions more reliable

*) *L. Buttyan and J.P. Hubaux*, “Security and Cooperation in Wireless Networks”

Existing Solutions



- EXAMPLE 1 "INTEL"
  - P2P Trusted Library based on `openssl`
  - allows for the establishment of "*trust*" between individual P2P clients and the organization of secure groups of "*trusted*" peers
  - C++ based API to implement security layer over P2P applications
  - *digital certificates, peer authentication, secure storage, public key encryption, digital signatures, and symmetric key encryption*
  - some operating system primitives, such as threads and locks



Existing Solutions

- EXAMPLE 2 ”JXTA“
  - open source P2P-protocol
  - security classes based on Java Card Security 2.1
  - *provides support for public key technology, symmetric key technology, hashing for authentication etc .*
  - provides security in terms of secure sockets and secure group authentication/privacy
  - *Algorithms like RC5 and SHA are supported.*
  - *ID := 160 bit SHA-1 URN in the Java binding*
  - *edge peers* and *super-peers (rendezvous- or relay peers)*
  - no routing security available
  - no overall security concept

Existing Solutions

- EXAMPLE 3 ".NET"
  - Framework provides a rich platform for building P2P apps
  - four application models
    - Web Services (`System.Web.Services`)
    - Windows Forms (`System.WinForms`)
    - Web Forms (`System.Web`)
    - Service Process (`System.ServiceProcess`)
  - *supports digital certificates, signatures,*
  - *hashing, random number*
  - *generation, asymmetric/symmetric*
  - *encryption, signing XML objects.*

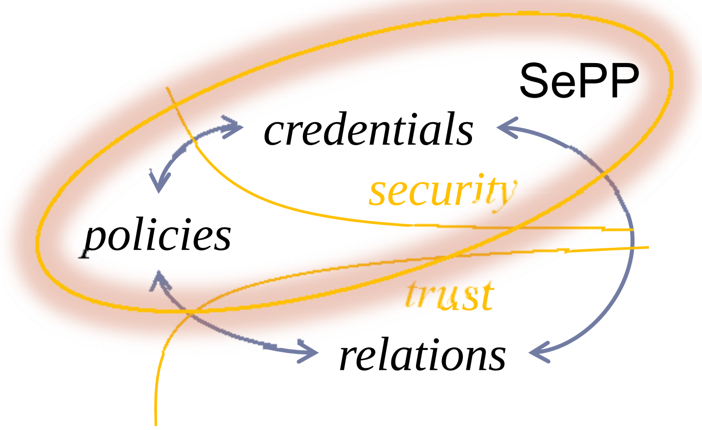
Existing Solutions

- EXAMPLE 4 " GROOVE"
  - Founded by Ray Ozzie in 1997 (*creator of Lotus Notes*) version 1.0 in April 2001
  - … now: Microsoft Office Groove 2007
  - Strong security – always on
  - Shared space data is confidential - no impersonation
  - No uninvited members can eavesdrop or temper group-data/info
  - Lost messages can be recovered from any member with assurance of integrity.

Source: http://office.microsoft.com/de-de/groove/default.aspx?ofcresset=1

Secure P2P (SePP) Concept

- Assumptions
  - *Most* of the authenticated *peers will be well-behaved*.
  - Attackers in possession of the credentials (*malicious insiders*) will be considered as an *exceptional case*.
    - *Must be detected by co-operating peers or network monitoring*
  - Attackers without proper credentials (*malicious outsiders*) *are restricted* to get access to the raw communication infrastructure (data streams).



Secure P2P (SePP) Concept

- The model would need mechanisms for ...
  - system creation
    - peer tries to find a suitable P2P system
      - e.g. decentralized pure P2P - they try to find peers in its reachable local area (*boadcasting*)
    - known addresses
    - specific multicast group
    - create its own system
  - peer admission
    - `joining` / `leaving` a group
    - guarantee the correctness of the participating entities (*identity)* by using *authentication*.
  - grouping mechanism
    - one global or general group exists from the beginning
    - new groups may contain peers of other groups which should be subgroups of the new group

Secure P2P (SePP) Concept

- Separated into 2 domains
  - *1.) Routing security* - every legitimate peer can `join` a routing group
  - *2.) Group security* - the set of peers (*able to join these groups*) can be restricted
- These aspects apply to both domains:
  - *Establishing*, *performing* and *upholding* secure communication within groups
- Underlying same "*basic*" group concept
  - A group is a virtual meeting place with membership requirements and available services
    - `join(), leave(), search(), create(), …`

Secure P2P (SePP) Concept

- **1.) Routing Security:**
  - Depending on the desired *level of security* it is possible that
    1. all peers can `join` without providing any credentials
    2. peers possessing a shared secret key can `join`
    3. peers with authorized public/private key pair can `join`
  - Currently supported: `http://sourceforge.net/projects/secureP2P`
    - DSR + secure variants Ariadne and SDSR
    - AODV + secure variants SAODV and AODV-S
      - Most preferable secure algorithm is Ariadne using TESLA for broadcast authentication
        - µTESLA has been applied to WSNs successfully
        - DSR is very simple and resource preserving

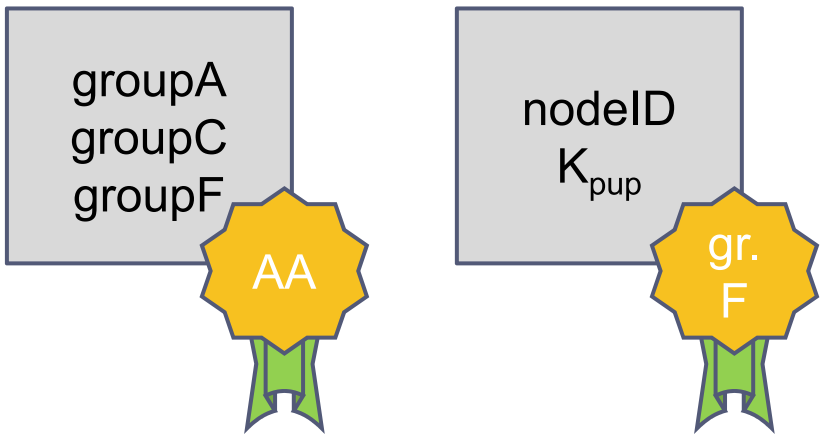
Secure P2P (SePP) Concept

- **2.) Group Security**
  - Assumptions (Entities)
    - Key Distribution Center (KDC)
      - We assume an offline KDC → obtain keying material before system startup.
    - Certificate Authority
      - We assume an offline CA → combine the KDC with the CA functionality.
    - Unique identity per peer
      - use the combined KDC/CA to create the identities and establish the key binding.

Secure P2P (SePP) Concept

2.) Group Security:

1. Admission Security
2. Data Security
3. Session Key Protection



- 1.) Admission security
  - pre-shared secret key (admission security *level 1*)
  - individual public/private key pair (*level 2*)
    - a single key pair with an attribute certificate (*of the public key*) containing the list of groups which can be joined
    - a separate key pair per peer for each group, where the *public keys for the same group are certified* with a different group public/private key pair
  - an additional dynamic group authorization service is possible but currently out of the scope of this simple model.

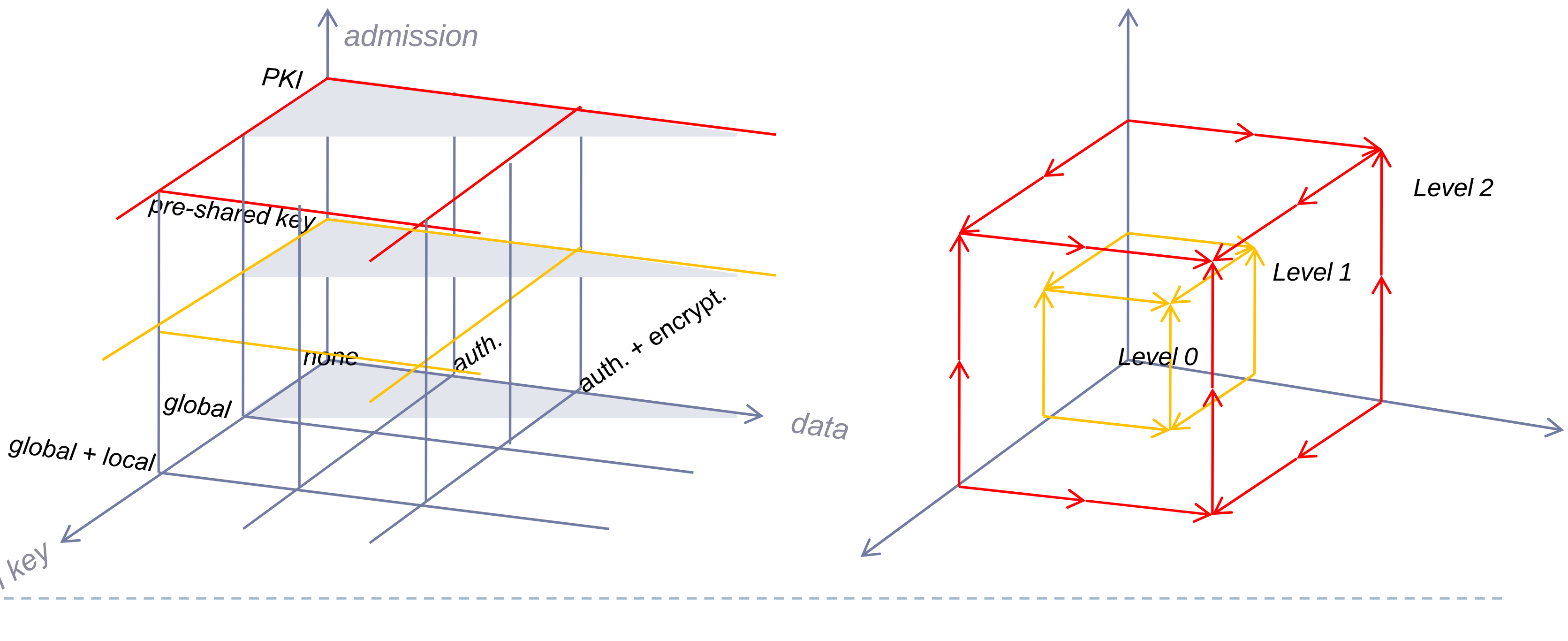
Secure P2P (SePP) Concept

2.) Group Security:

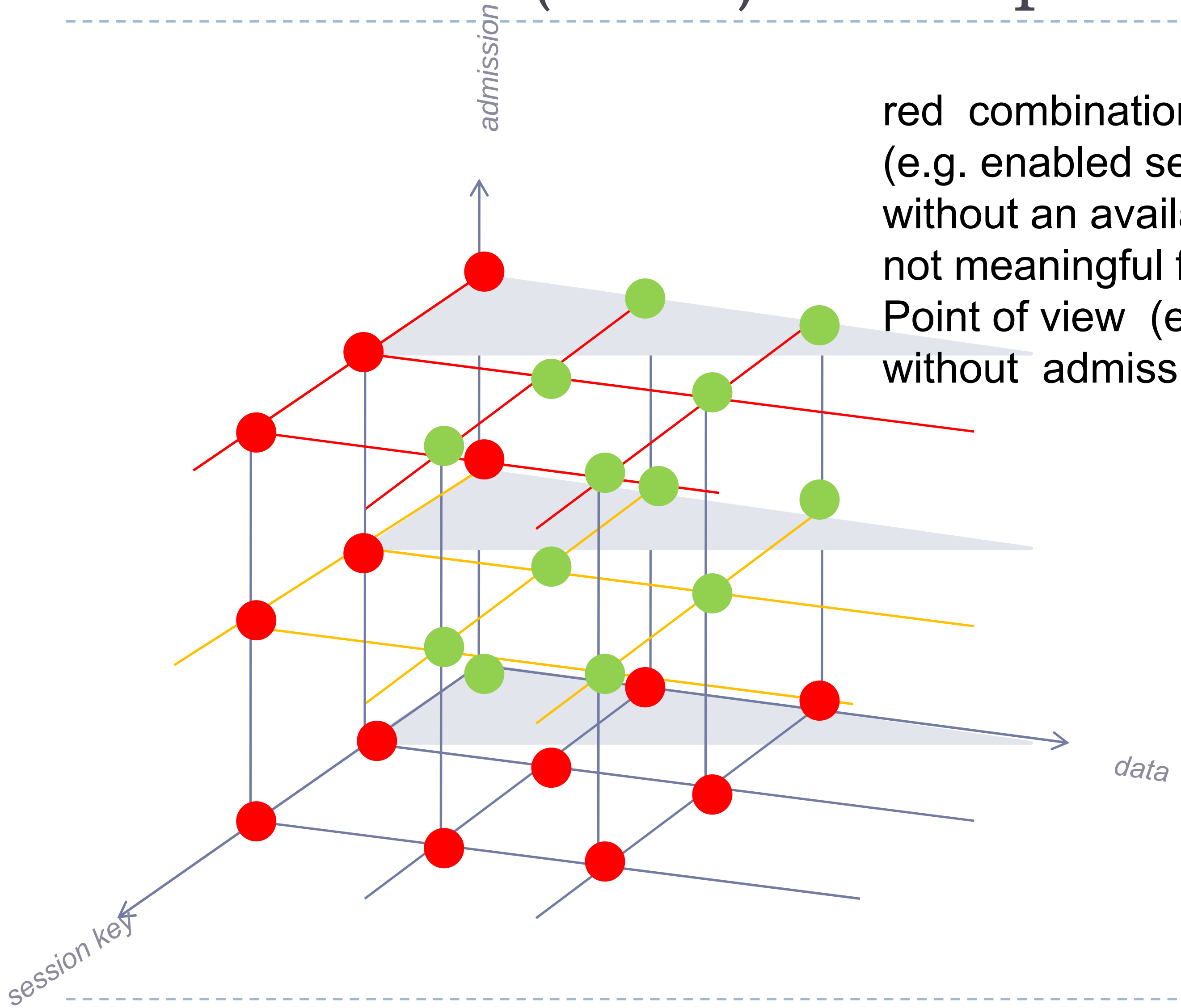
- 2.) Data security
  - for *routing* and *data*
  - data must be at least *authenticated* (*authentic peer or group member*) → level 1.
  - data must be *encrypted* → level 2.
- 3.) Session key protection
  - hardening the extraction of session keys from devices
  - limiting the value of session keys learned by an attacker
  - *periodic refreshing decreases the chance of successful side-channel attacks*
    - never refreshed→ level 0.
    - key is refreshed at certain intervals → level 1.
    - ? → level ?
    - side-channel resistant implementations → level 2.

Secure P2P (SePP) Concept

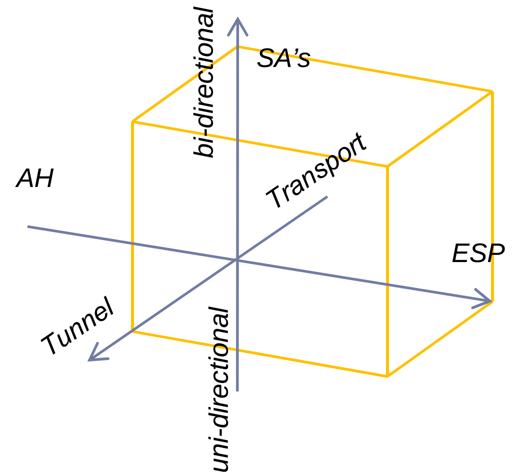
- Security levels
  - admission security (entity authentication and authorization)
  - data security (message authentication and confidentiality)
  - session key protection (global re-keying and local side-channel attack countermeasures)



Secure P2P (SePP) Concept



red combinations are either invalid (e.g. enabled session key protection without an available session key) or not meaningful from a security Point of view (e.g. enabled data security without admission security).



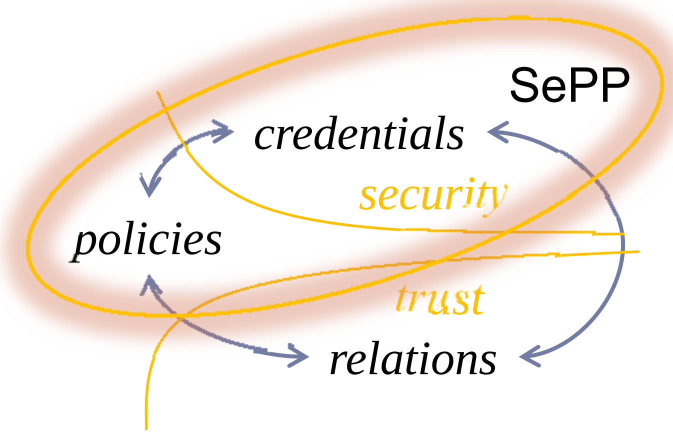
Schneier/Ferguson: „A Cryptographic Evaluation of IPsec"

Secure P2P (SePP) Concept

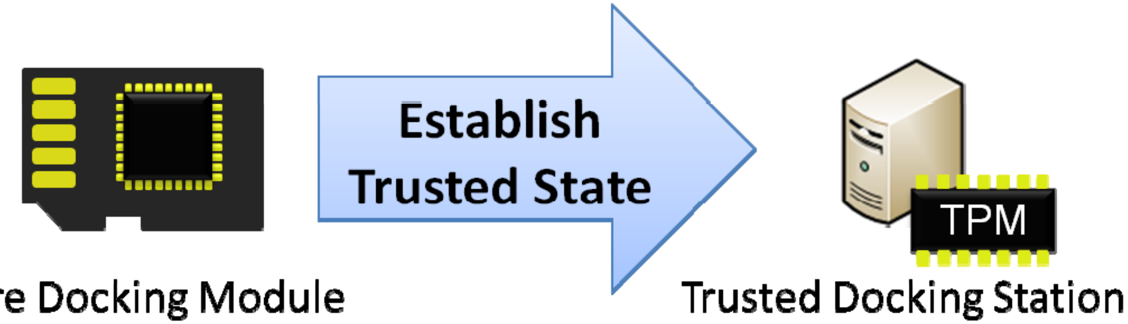
- Selection of security levels
  - The routing security level is a global decision.
    - → admission security level and data security level for routing must be set globally
    - ..., as well as session key protection level
  - Group security levels are set group-wise
    - decided by the *group creator*
    - *admission security level* and *data security level* as well as the *session key protection* level

Conclusion

- Nice "*trust-*" models exist (*not only academic models*)
- ... but Blaze Trust Model *is not implemented (so far)*
- SePP = first step towards a general P2P security concept
- Open issues:
  - Coarse grained approach needs to be refined
  - several routing algorithms are available – needs to be completed (incorporating the security concept)
  - ... how to integrate trust?



Security-Trust Binding



key id ← — key request

create nonce n

n →

generate platform configuration report (quote)

$TPM_{sig}(n,quote)$ ←

Verify signature, nonce, quote

key → — receive key

Security-Trust Binding