Introduction to IT Security

Summer School 2015 on Smart Energy Systems & Entrepreneurship
EIT ICT Labs & KIT. Karlsruhe, Germany. July 30th, 2015

Dr. David von Oheimb, Siemens AG, Corporate Technology

SIEMENS

What is IT Security?

- What is security about? – It‘s about bad things not happening.
- Where is the *difference to safety?* – It’s in the source of bad things.
- *Safety* protects against accidents due to technical failure or human mistakes.
- *Security* protects against evil due to **malicious human intentions**.
- Which is harder to achieve? – Security.
- Why? – Because people can be very creative and determined to search for and exploit vulnerabilities, while safety failures happen by chance.
- Is it simpler to attack or defend? – Attack.
- Why? – Only one weak point is needed to break in.
- What is *IT* security? – **Protection of data against unauthorized access.** *Not:*

SIEMENS

IT Security may impact safety – Example: Boeing 787

- **Situation**
  - Aircraft flight is controlled by avionics
  - Malfunction may lead to catastrophic accidents
  - Today's avionics is software controlled
  - Sabotage of software may cause malfunction
  - Avionics software can be updated via networks
  - Transmission of software might be attacked
  - *Classical IT security threatens flight safety!*

Measures

- *Boeing R&D* developed BEDS (Boeing Electronic Distribution of Software)
- *Siemens CT* assisted Boeing R&D in analyzing the security threats, designing proper countermeasures, and defining certification approach

SIEMENS

Threat Agents and Their Motivation

- How does one call people threatening IT security?
- Commonly they are called *hackers*.
- Security folks speak of *attackers*.
- Researchers tend to call them *adversaries*.

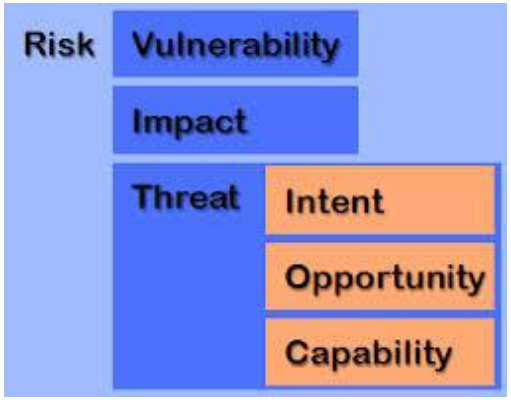
- What is the aim of attackers?
- *Script kiddies* typically want to have fun.
- *Criminals* typically want to steal money.
- *Insider attackers* typically want to take revenge.
- *Political activists* want to control decisions.
- *Terrorists* want to threaten society.
- *Spies* want to gain (technical/economic/organizational) knowledge.
- *Secret services* want to gain knowledge and influence at large scale.

SIEMENS

Vulnerabilities, Threats, Risk, and Aim of Security

- A vulnerability is a weakness (loophole) that could be used to do harm.
- An exploit is the use of a vulnerability for performing an attack.
- *An attack is* an activity exploiting vulnerabilities, typically to do harm.
- *Attack potential* is the strength of an attack (amount of ability, energy, motivation).
- *Impact* is the amount of harm to assets achieved via a successful attack.
- *Risk* is the amount of damage to be expected:

  Probability of successful attack × Impact



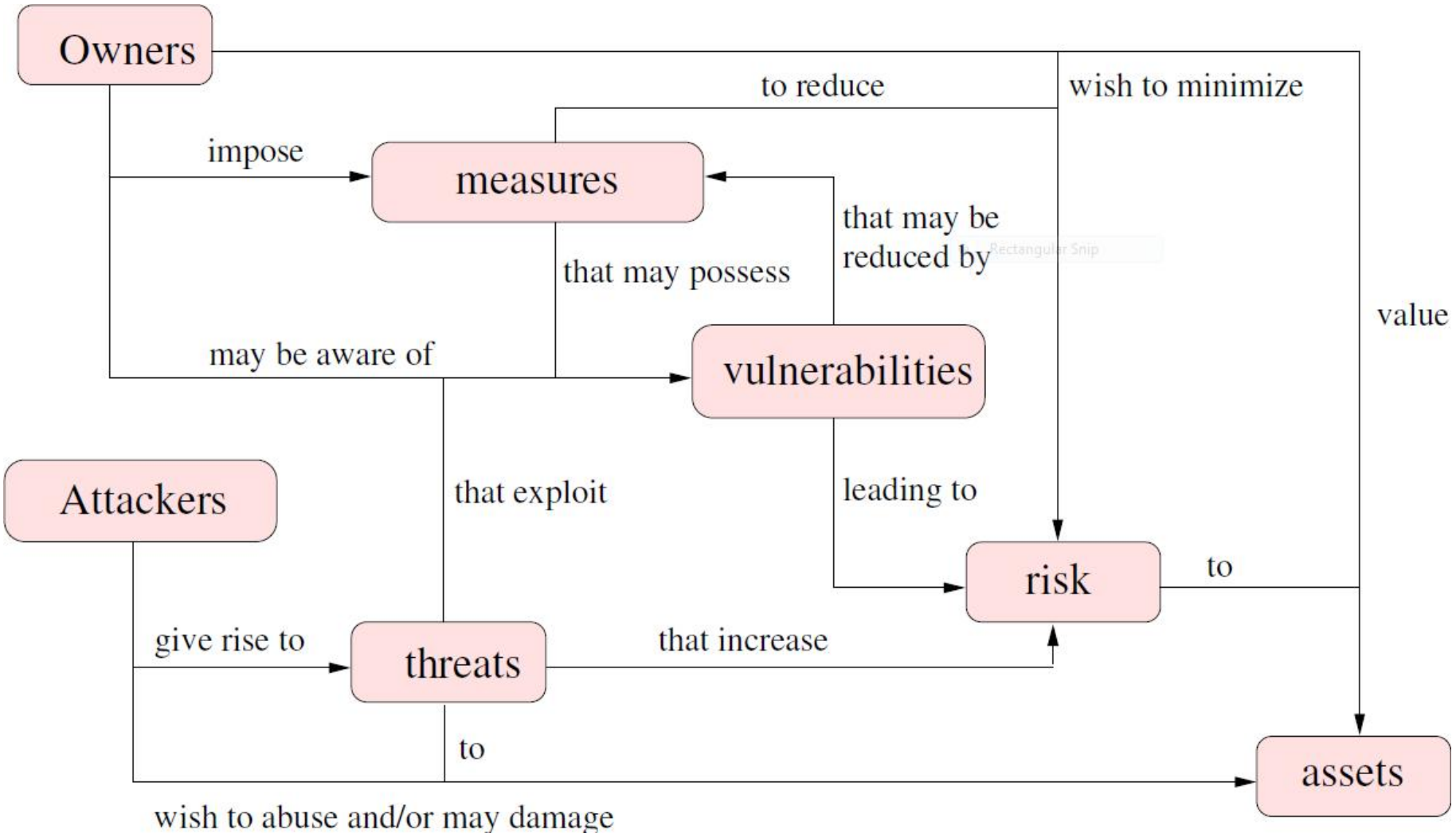
- Security aims at minimizing risk, by

  limiting potential impact, or better by

  minimizing vulnerabilities and/or opportunity,
  thus lowering the probability of successful attacks.
- **Residual risk is unavoidable – there is no 100% security!**

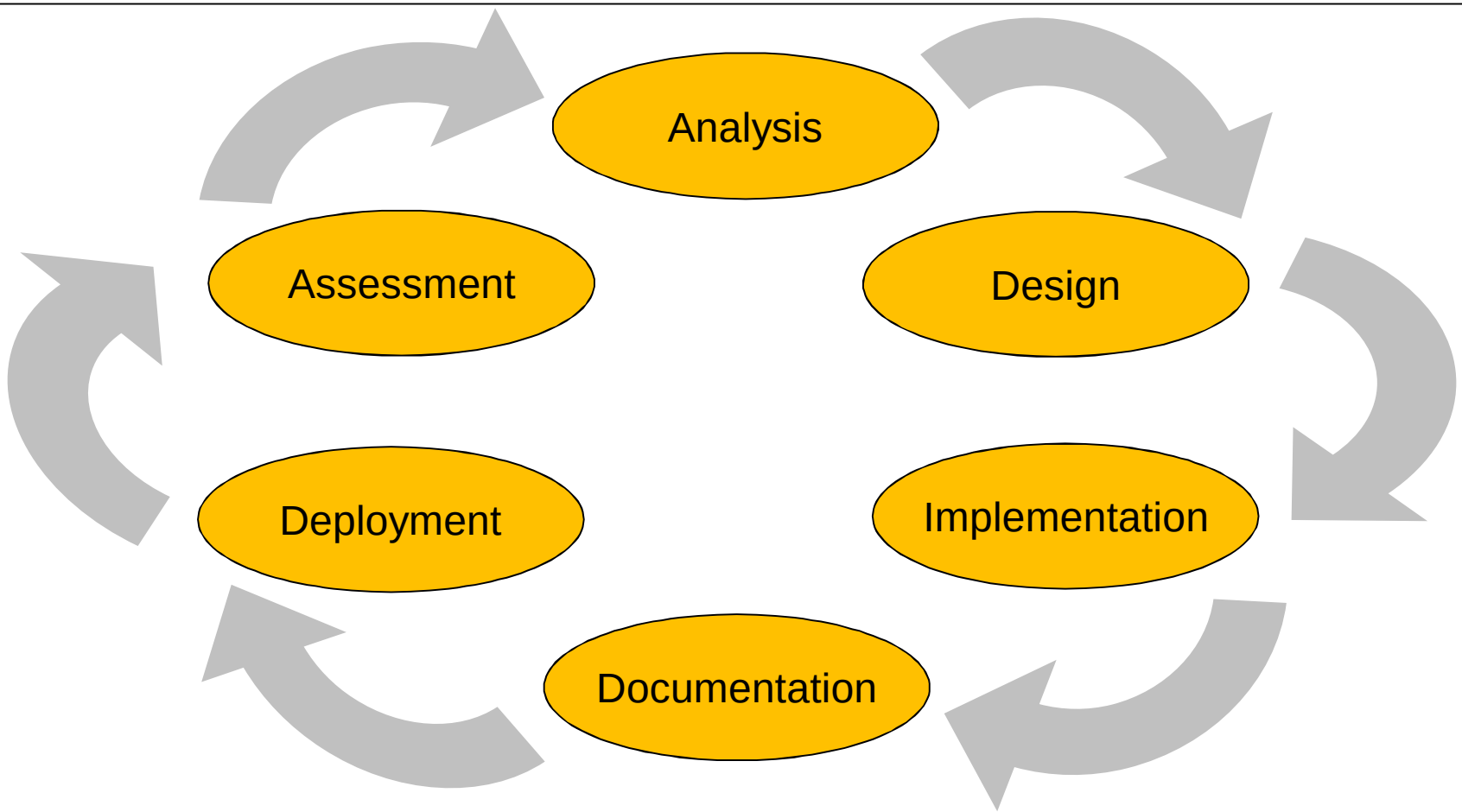
SIEMENS

Assets protected by owners and threatened by attackers



SIEMENS

Process to assure holistic security

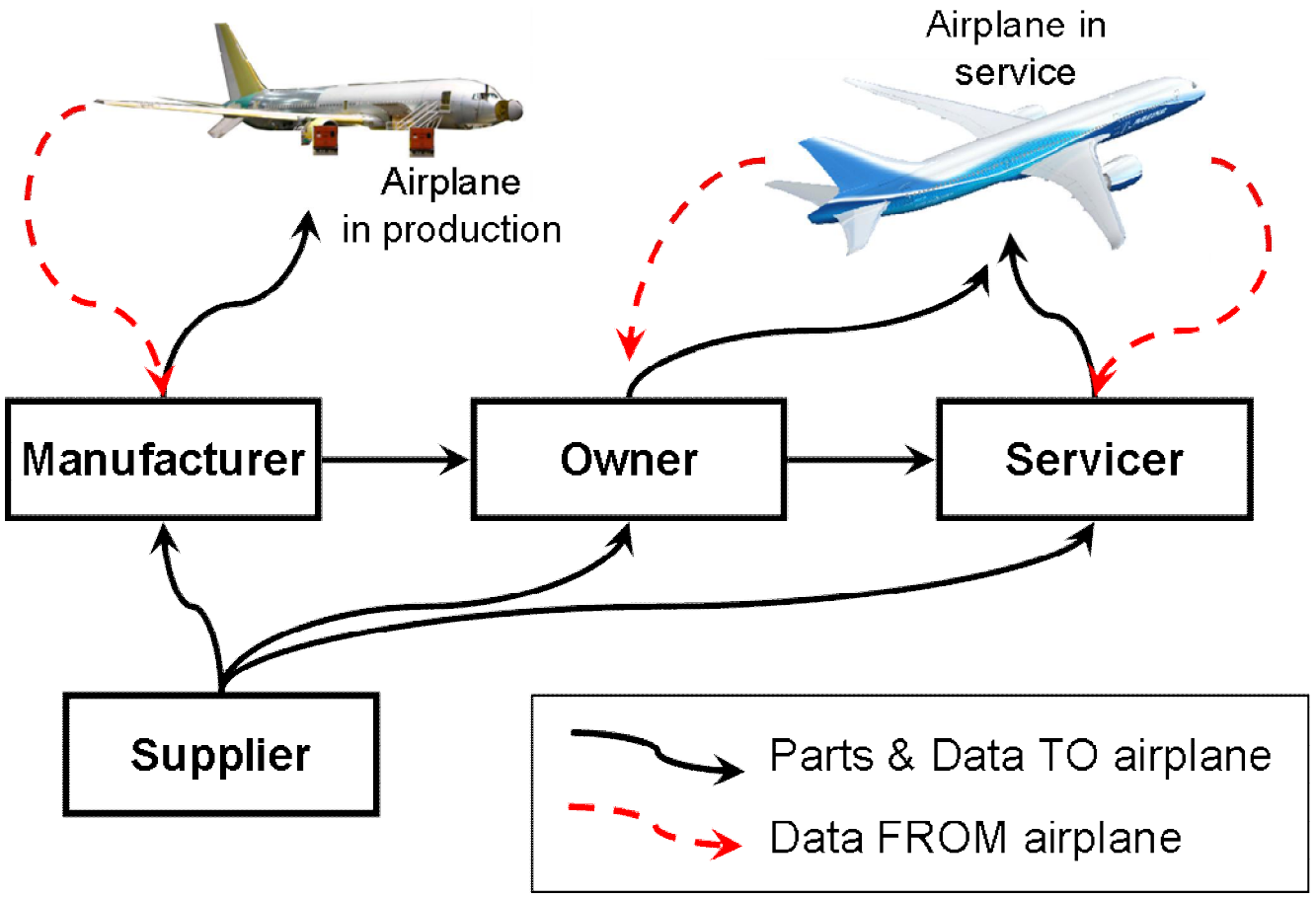


Any of these activities can be chosen as entry point.
In each of them, mistakes easily lead to an insecure system.

SIEMENS

Analysis step 1: Know your system and its assets

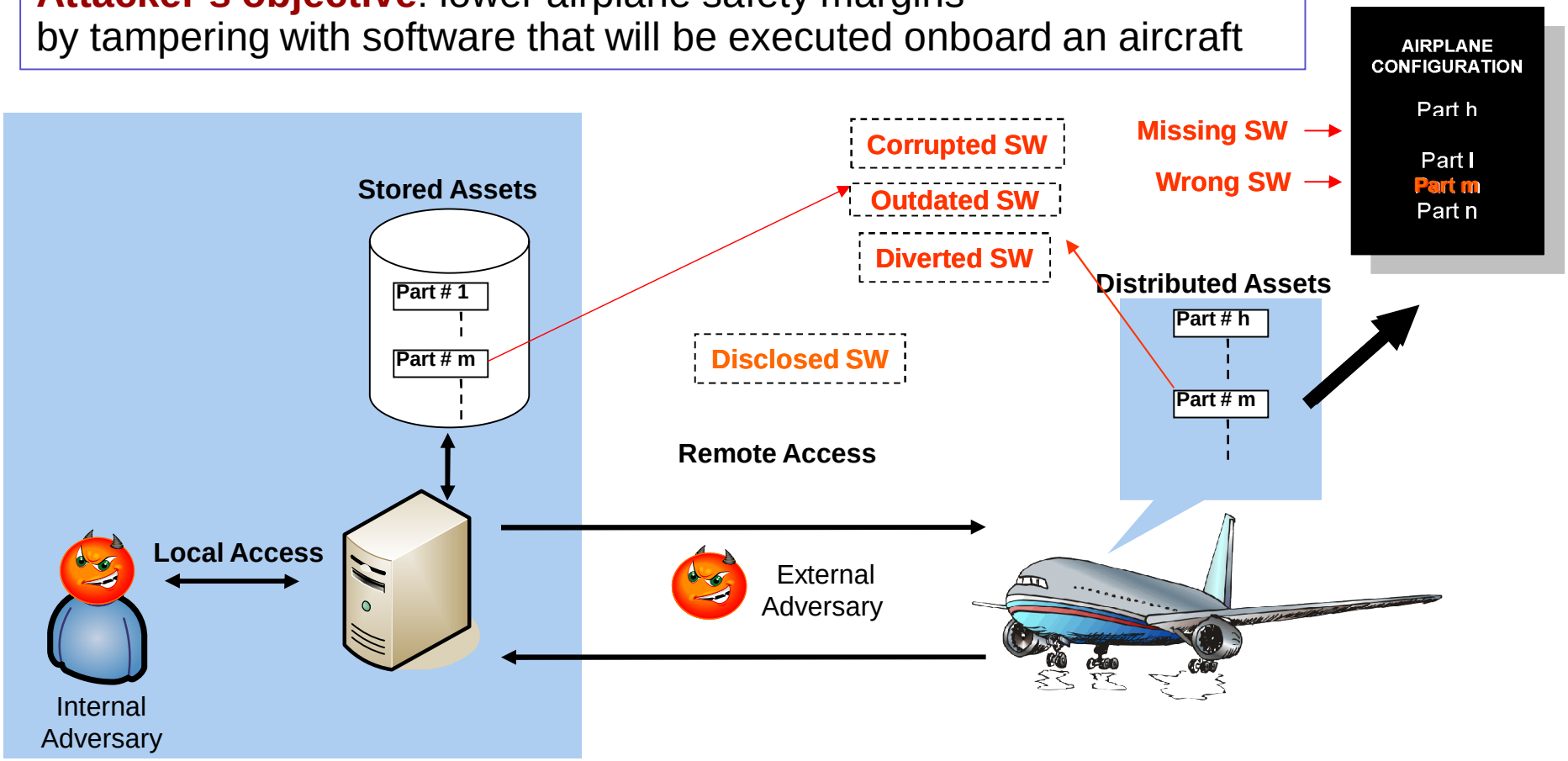
BEDS is a system for storage and distribution of airplane assets, including *Loadable Software Airplane Parts* (LSAP) and airplane health data



SIEMENS

Analysis Step 2: Know your enemy and the threats to your assets

Attacker's objective: lower airplane safety margins
by tampering with software that will be executed onboard an aircraft



Corruption/Injection **Wrong Version** **Diversion** **Disclosure**

Security Goals

IT security aims at protecting data assets against threats. Classical goals:

C **Confidentiality**: Information must be disclosed to certain parties only

Example: personal information to be kept secret

I **Integrity & Authenticity**: Information must be changed by certain parties only; fake or manipulation must be detectable

Example: Signed contract must not be alterable

A **Availability**: Access by legitimate parties to asset must not be blocked

Example: Web server should remain usable.

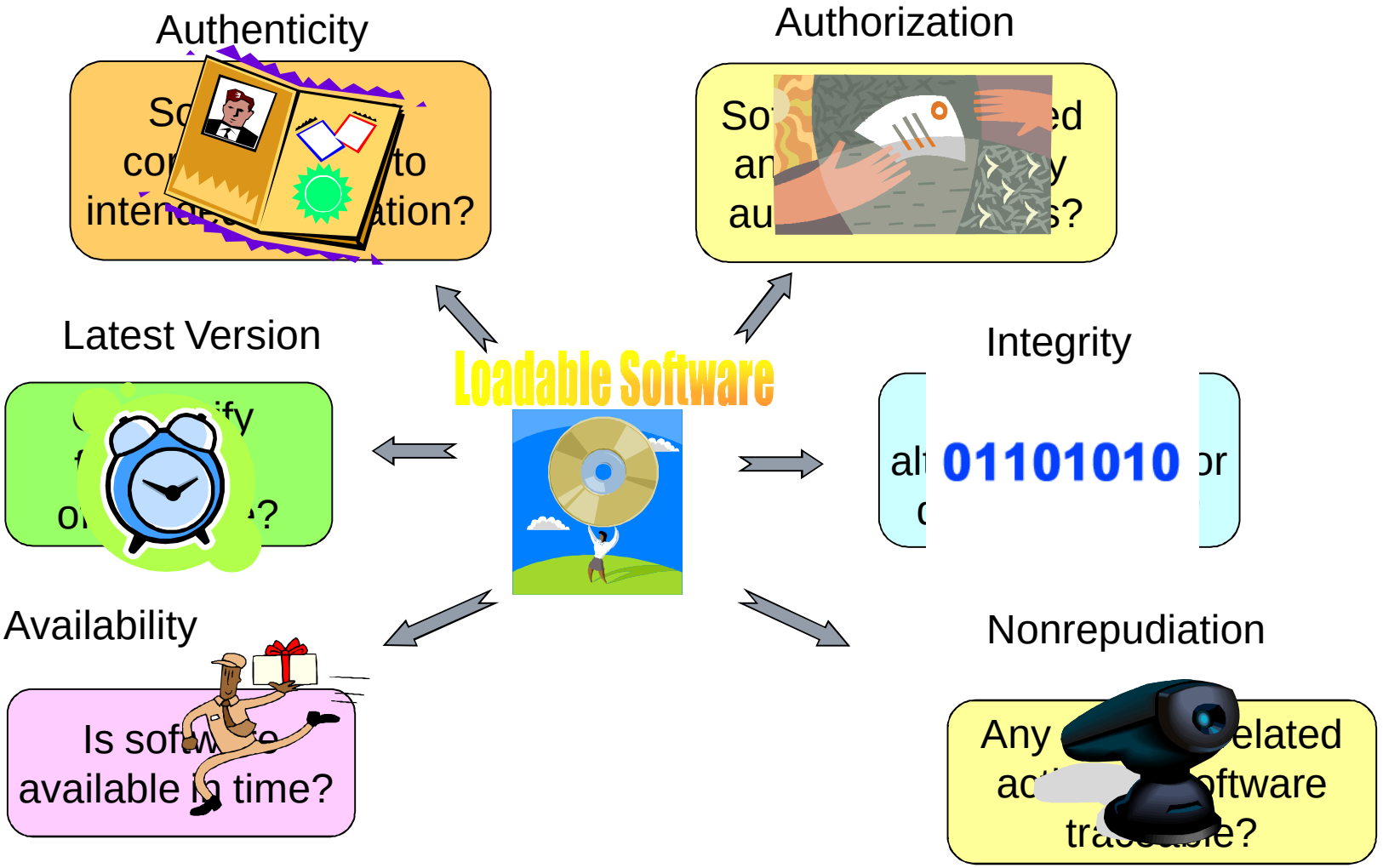
Further ones exist, e.g.:

Authorization: Only privileged parties must be able to perform action.

Non-Repudiation: Parties must not be able to deny certain actions.

Example: Buyer is accountable for accepted deal.

Design Step 1: Define security objectives



Security Policies and Mechanisms

A **security policy** describes in sufficient detail who is allowed to do what.

Everything not allowed is considered an attack and must be prevented.

How to protect against illicit access?

Direct/simple approach: Use a gatekeeper to **enforce the policy**.

Example: File access control by operating systems

Problem: The power/domain of the gatekeeper doing access control is limited.

Example: The operating system cannot protect while not running.

Advanced approach: Use cryptography such that access requires knowledge.

Example 1: Encrypt a secret with a key known only to the group allowed to know the secret.

Example 2: Digitally sign a document such that nobody else can fake the signature but everybody can verify it.

SIEMENS

Symmetric Cryptography

Any number of parties (e.g., A and B) share a key K.

A encrypts secret X with K, sends it, B can decrypt it: $A \rightarrow B: \{X\}_K$

Two problems: How to distribute the shared key without revealing it?

Idea: Use other channel protected in different way.
A better solution will follow.

How to make sure that K cannot be guessed/tried out?

Use sufficiently long key and good random number generator.

Algorithms: DES – outdated, key length 56 bits

AES – current, key length 128 or 256 bits

Hashing

Can one use symmetric encryption also for integrity protection for data Y?

Yes, use a checksum h() of K and Y, and add it: A → B: Y.h(K.Y)

Two caveats: Make sure that h is not invertible,
i.e., one cannot deduce X from knowing h(X).

Make sure that h is pre-image resistant, i.e.,
different X and X' should lead to different h(X) and h(X').

C**ryptographically strong checksums** are called **hashes**.

Algorithms: MD5 – outdated, result length 128 bits

SHA – current, result lengths 128 or 256 bits

General problem with symmetric integrity protection (aka HMAC)?
Everyone knowing K can produce h(K.Y) – this is not good as a signature.

SIEMENS

Asymmetric Cryptography

Each party A has a pair of a private key priv(A) and public key pk(A).

The public key can be known to everyone, while
the private key must not be shared with anyone.

If A sends secret X encrypted with pk(B),
only B can decrypt it using his corresponding private key: $A \rightarrow B: \{X\}_{pk(B)}$

This partly solves the key distribution problem: A can freely access pk(B).

Yet one problem remains: the authenticity of pk(B).

Asymmetric cryptography allows for real **digital signatures**:

If A sends Y with its hash encrypted with priv(A), written $A \rightarrow B: \{Y\}_{priv(A)}$
everyone can verify it using pk(A), but nobody can fake it!

Algorithms: RSA – key length 1024 or 2048 or 4096 bits

ECC – key length 80 or 112 or 160 bits has same strength

SIEMENS

Digital Certificates and PKI

Signatures can be used also for solving the authenticity the public key of B:

A third party C signs a **certificate** with name B and pk(B): $\{B.pk(B)\}_{priv(C)}$

Such a trusted third party is called a **Certificate Authority (CA).**

For sending secrets to B or verifying a signature of B, use pk(B), where its authenticity is verified using B's certificate.

To verify in turn B's certificate, pk(C) must be trusted or itself be verified using a certificate for C, issued by another CA, until reaching a trusted root CA.

Typically, certificates have a validity period and possibly further attributes.

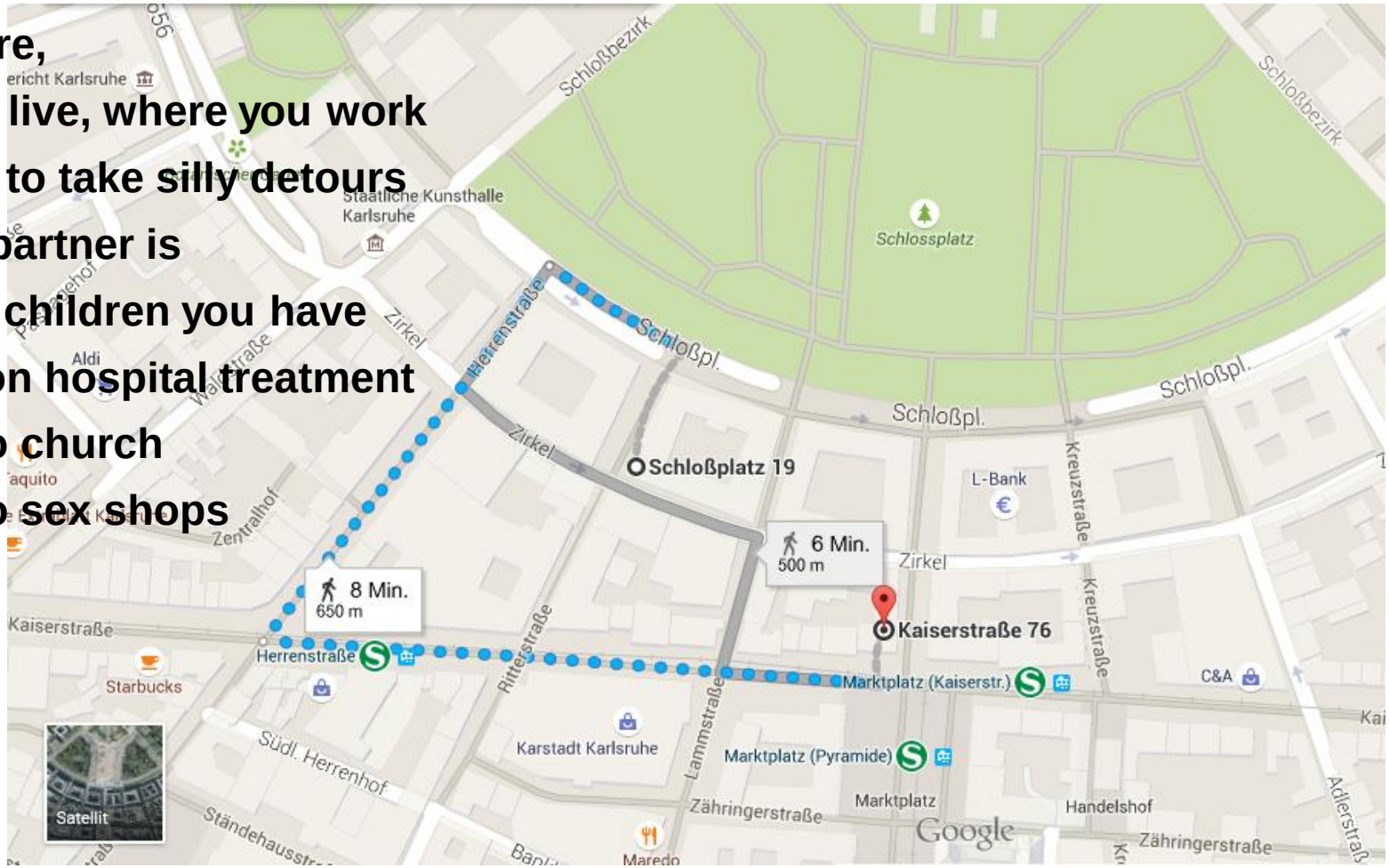
Standards: X.509 – defines format of large variety of attributes

Certificates need to be regularly updated and might be revoked, this their status is non-trivial to check.

The set of services handling all this is called **Public-Key Infrastructure (PKI)**.

Privacy is Tricky. Example: Location Data, e.g., From Smartphone

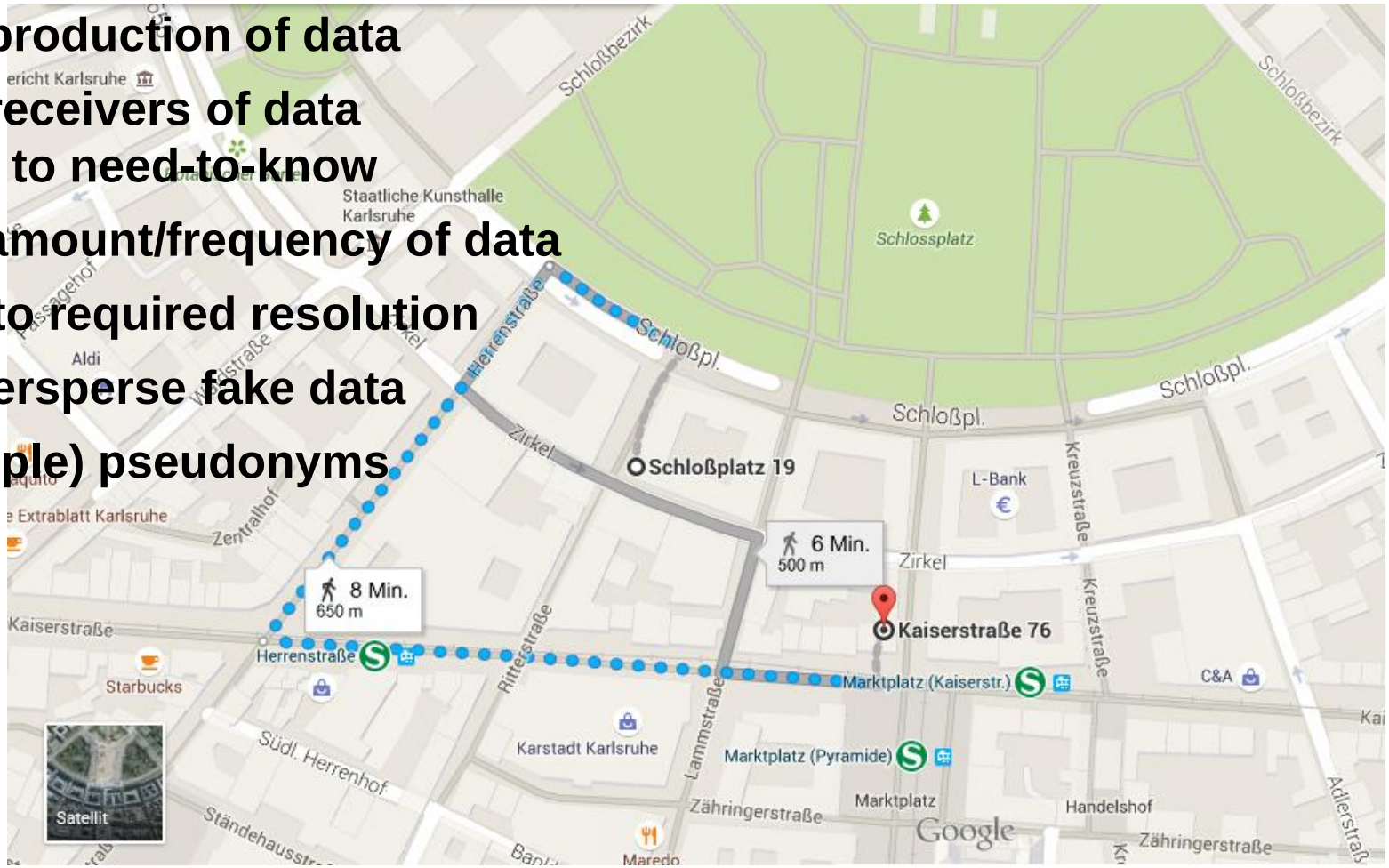
- If I have your location data over some time, I can tell:
- **Who you are,**
- **Where you live, where you work**
- **If you tend to take silly detours**
- **Who your partner is**
- **How many children you have**
- **If you are on hospital treatment**
- **If you go to church**
- **If you to go sex shops**



SIEMENS

How to support privacy

- Principle of data sparingness:
  Only give as much data as needed, to as little parties as possible.
- **Minimize production of data**
- **Minimize receivers of data according to need-to-know**
- **Limit the amount/frequency of data**
- **Blur data to required resolution**
- **Maybe intersperse fake data**
- **Use (multiple) pseudonyms**



SIEMENS

Frequent Security Mistakes

What if a **programming error** anywhere in SW leads to a severe vulerability?

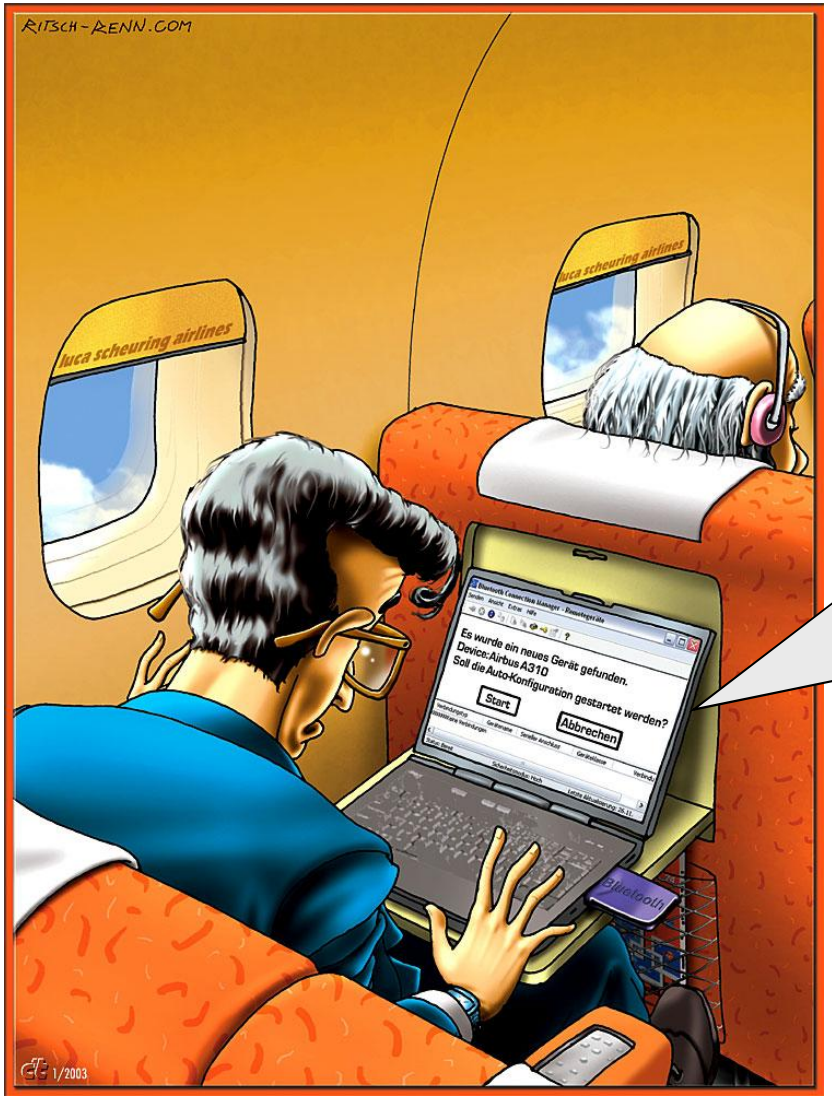
- Many such examples exist: buffer overflows, missing input validation, …

Why employ the best programmers if you **forgot an important requirement**?

What is any security mechanism worth if it can be **circumvented**?

- For instance, critical web server data may be accessible without login.

What Should Not Happen ;-)



A new device was found.

Device: Airbus A 310.

Shall auto-configuration be started?

start **cancel**

SIEMENS

Frequent Security Mistakes

What if a **programming error** anywhere in SW leads to a severe vulerability?

- Many such examples exist: buffer overflows, missing input validation, …

Why employ the best programmers if you **forgot an important requirement**?

What is any security mechanism worth if it can be **circumvented**?

- For instance, critical web server data may be accessible without login.

What does a decent security mechanism help if is **wrongly implemented**?

- Suppose the signature function uses the same test key on all systems.

Why use a strongest crypto algorithm if **secret keys can be leaked**?

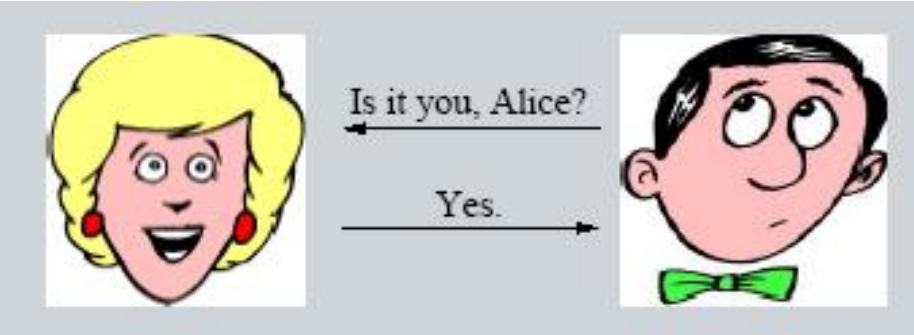
- E.g, due to differential power analysis (DPA)

What if your security mechanism has a **design flaw**?

- See Needham-Schroder Protocol example

Needham-Schroeder Public Key Protocol

- [Needham & Schroeder 1978]
- http://en.wikipedia.org/wiki/Needham-Schroeder_protocol
- Goal: strong mutual authentication



- Simplified version without key server, assuming that A and B already know the public key of their peers:

- $A \rightarrow B: \{Na.A\}_{pk(B)}$
- $B \rightarrow A: \{Na.Nb\}_{pk(A)}$
- $A \rightarrow B: \{Nb\}_{pk(B)}$

- Suffers from Man-In-The-Middle attack!

SIEMENS

Lowe's attack on NSPK

- [Lowe 1995] Man-in-the-middle attack by dishonest peer of A
- Requires two interleaved sessions, each with one honest party.
- In the first session, Alice talks with some party, e.g. Chuck, who in fact is an adversary, also called intruder.

1.1	A	- $\{Na.A\}_{pk(C)}$ ->		C
2.1			C(A) - $\{Na.A\}_{pk(B)}$ ->	B
2.2			C(A) <- $\{Na.Nb\}_{pk(A)}$ -	B
1.2	A <- $\{Na.Nb\}_{pk(A)}$ -			C
1.3	A	- $\{Nb\}_{pk(C)}$ --->		C
2.3			C(A) - $\{Nb\}_{pk(B)}$ -->	B

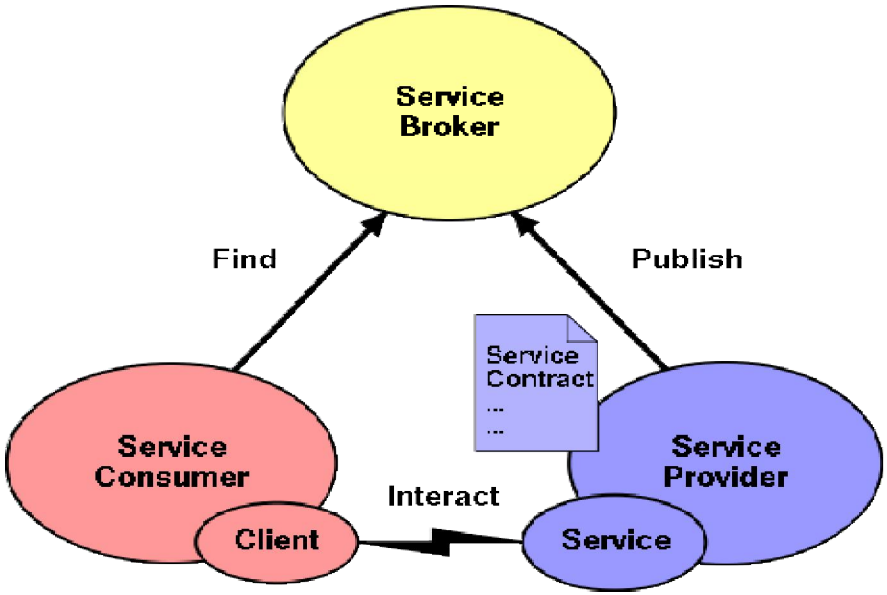
- In the second session, Bob thinks that he was contacted by Alice but actually talks to A via the intruder.
- Therefore, anything echoed by A, like Nb, gets leaked to the intruder.
- The protocol can be fixed by adding B's name to the 2nd message: $\{Na.Nb.B\}_{pk(A)}$

AVANTSSAR

SIEMENS

avantssar.eu

Model-checking SOA security — research project AVANTSSAR[1]



[1] **A**utomated **Va**lidatio**N** of **T**rust and **S**ecurity of **S**ervice-oriented **Ar**chitectures

FP7-2007-ICT-1, ICT-1.1.4, STREP project no. 216471
Jan 2008 - Dec 2010, 590 PMs, 6M€ budget, 3.8M€ EC contribution

SIEMENS

How Not To Do Security ;-)