ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

inf | Informatik
Computer Science

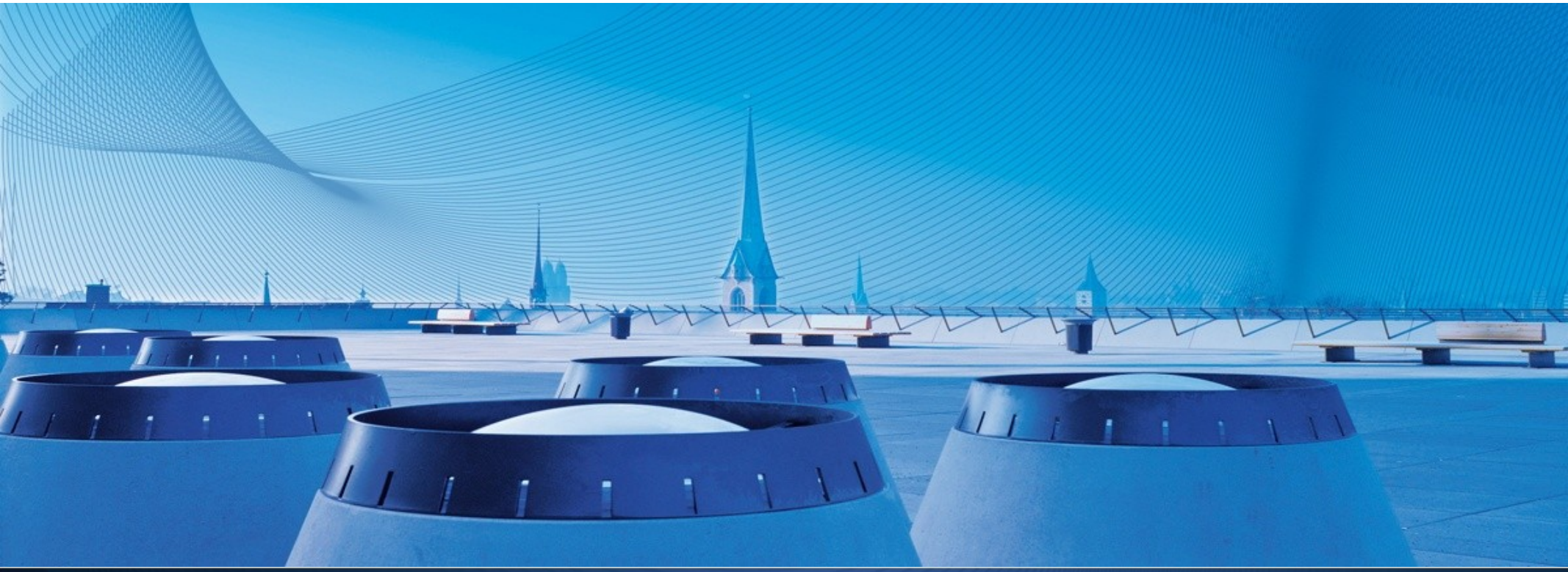
Secure Platform Problem

- Current Mitigation Approaches -

E-Voting Seminar

24 June 2011

Michael Schläpfer



Good crypto alone … remember?

Security protocols

A **protocol** consists of a set of rules (conventions) that determine the exchange of messages between two or more principals. In short, a **distributed algorithm** with emphasis on communication.

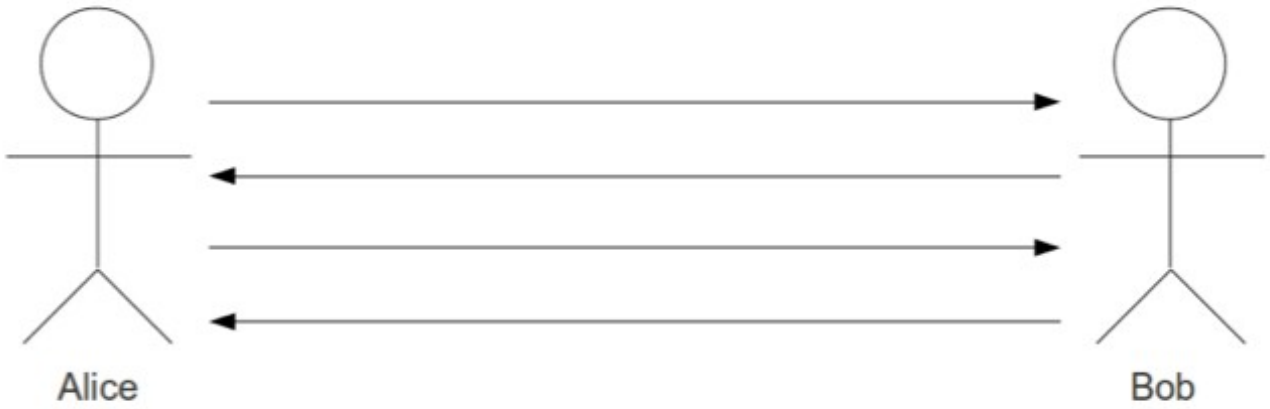
Security (or **cryptographic**) protocols use cryptographic mechanisms to achieve security objectives.

Some common security objectives:

- Entity or message authentication
- Key establishment
- Integrity
- Fair exchange
- Non-repudiation
- ...

Motivation

Communication in an ideal world.



Does this reflect the Internet?

Motivation

Communication in an dangerous world.

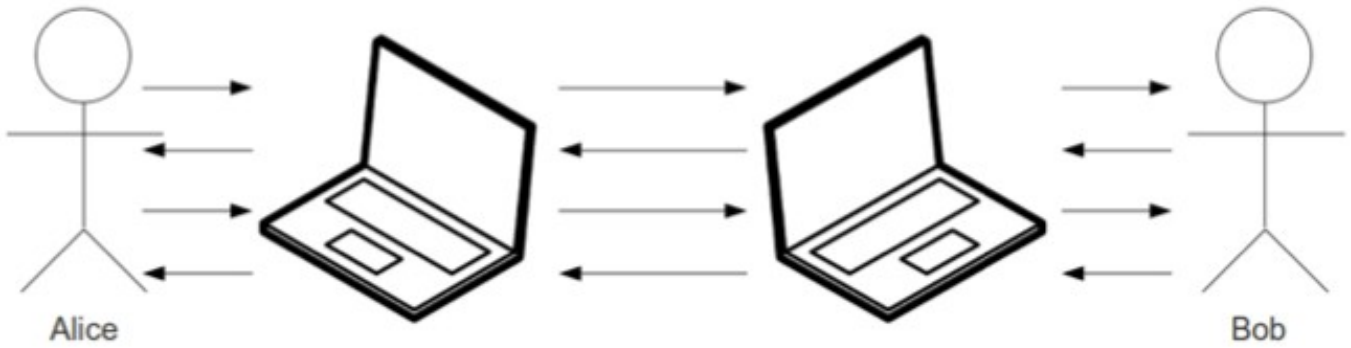
- **On the Security of Public Key Protocols (IEEE Trans. Inf. Th. 1983):**
- Danny Dolev
- Andrew C. Yao

The Dolev-Yao Intruder:

- Controls the network (read, intercept, send)
- Is a legitimate user
- Can apply every publicly available information or function
- Can apply his private information and functions
- Cannot break cryptography

Motivation

A more realistic setting for electronic communication.

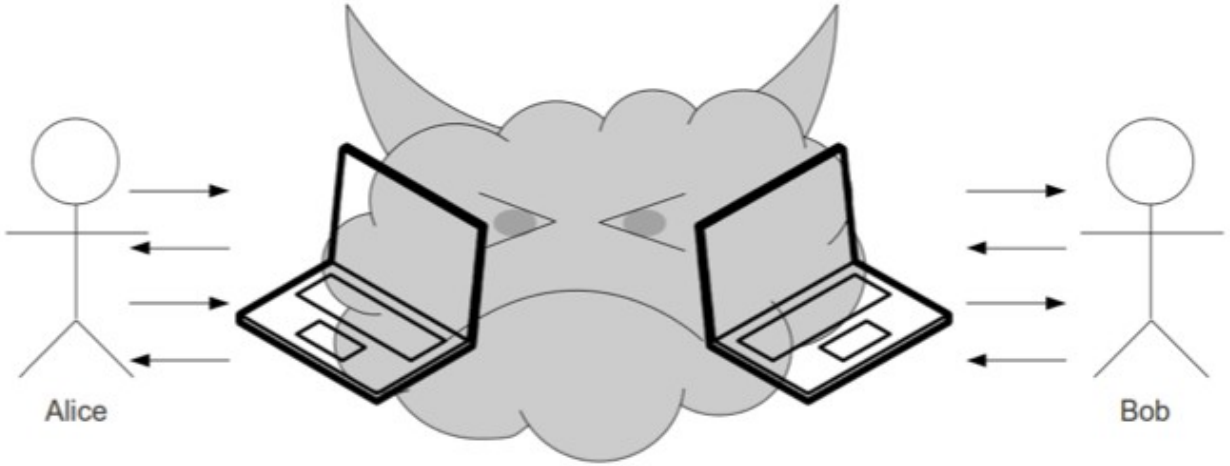


Is the attacker restricted to controlling the network only?

- More and more complex Operating Systems
- Growing number of codes of lines leads to a growing number of vulnerabilities
- Increasing number of sophisticated malware
- Most dangerous malware uses exploits that are not yet publicly known!

Motivation

A more realistic view on the attacker.

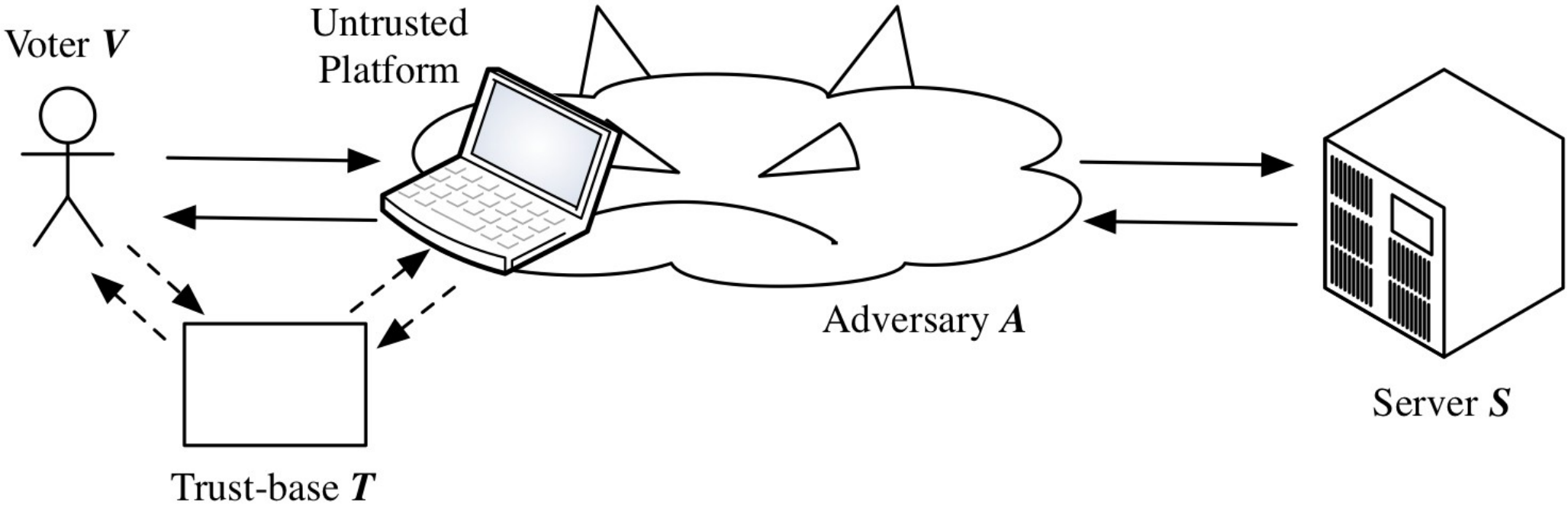


Depending on the application and the resulting threat sources we will have to assume a very powerful attacker, capable of controlling the entire computing platform.

So we can say: good protocols alone ...

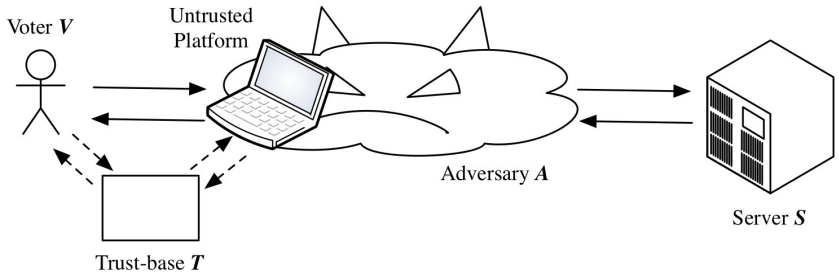
Overview

Problem Description (In E-Voting)



- We abstract from the voter's platform
- The attacker offers the network services to the voter
- Abstracting from the construction of the messages as it is done in Dolev-Yao-like models cannot cover the Secure Platform Problem
- Trusted functionalities modeled by a trust-base

Modelling the attacker



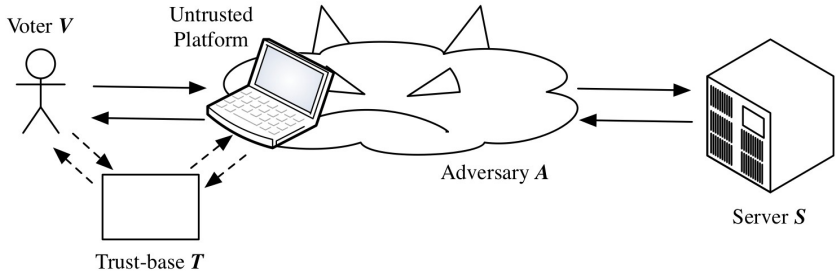
The attacker's capabilities:

- he can learn messages sent from V to her untrusted platform that is part of A
- he can learn messages sent from S to the network and further to an untrusted platform, both parts of A
- he can learn messages sent from T to an untrusted platform or directly to the network, both parts of A
- he can drop messages to exchange them with own messages
- he can manipulate and fabricate arbitrary messages according to his and publicly known knowledge
- he can perform every publicly known function
- he has human capabilities too and therefore can solve CAPTCHAs
- he knows all implementation details of all used systems

The attacker's limitations:

- he is computationally bounded and thus cannot break cryptography
- he cannot overhear or manipulate the communication between V and T
- he cannot access knowledge dedicated to V, T, or S
- he cannot exchange T or parts of it

Security Properties



Secrecy:
An approach supports V-to-S-secrecy if it is not possible for A to learn a secret that V submitted to S. This means that the voter V has the opportunity to submit her choice secretly to the authority's server S and hence voter privacy holds.

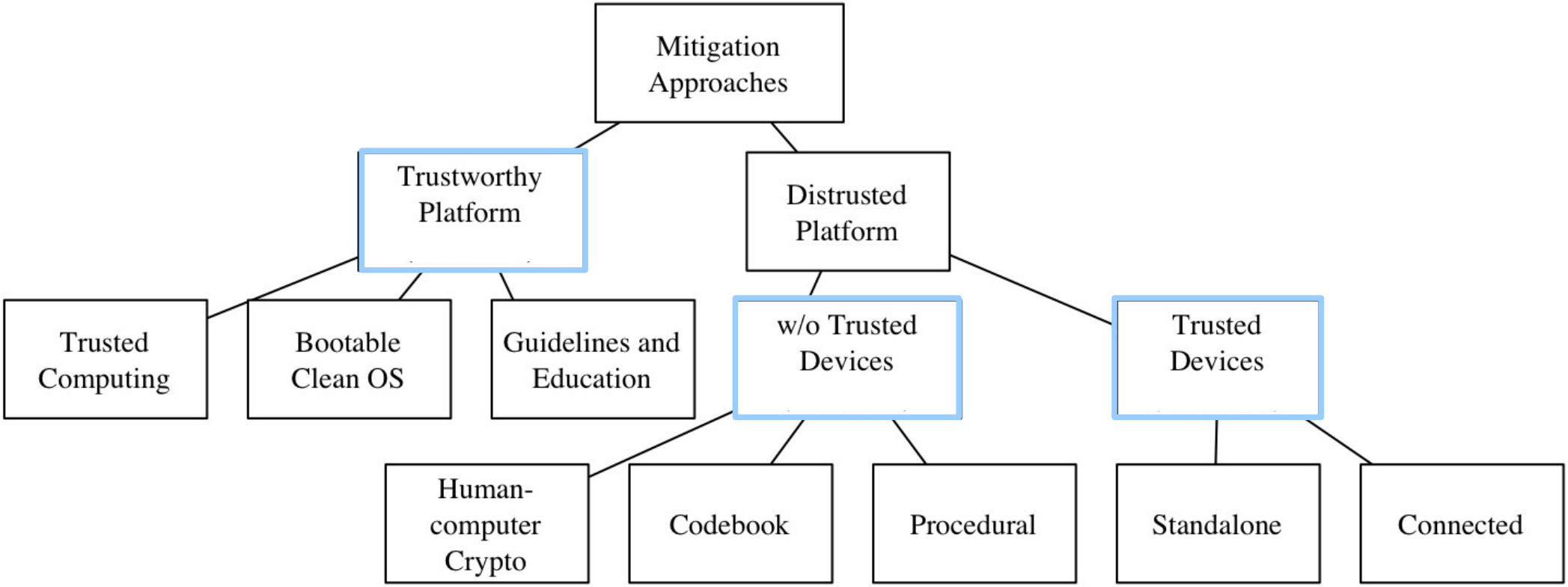
Anonymity:
It is possible for V to send messages that are not secret but cannot be linked to V and thus voter privacy holds too. We say an approach supports V-anonymity if it is not possible for A to reveal the origin of a message sent to S. Note that fairness may not hold if A learns anonymous votes during the vote casting phase.

Integrity:
If it is possible for S to verify the integrity of a received message with respect to V's choice, we say the approach supports V-to-S-integrity, which is vote integrity. Note that individual verif ability allows the voter to verify that S correctly received the voter's choice. But this is not enough to ensure vote integrity since the voter must also be given the possibility to complain in the case the verif cation fails. Moreover, individual verif ability requires also integrity of messages sent from S to V in order for V to be able to verify what S received. From the authority's perspective vote integrity is then given by the fact that the voter has not complained.

Overview

1. Motivation
2. Problem Description
3. Taxonomy of Mitigation Approaches
4. Making the Platform Trustworthy
5. Distrusting the Platform (Approaches w/o Additional Devices)
6. Distrusting the Platform (Personal Trusted Devices)
7. Conclusion and Future Work

Taxonomy of Mitigation Approaches

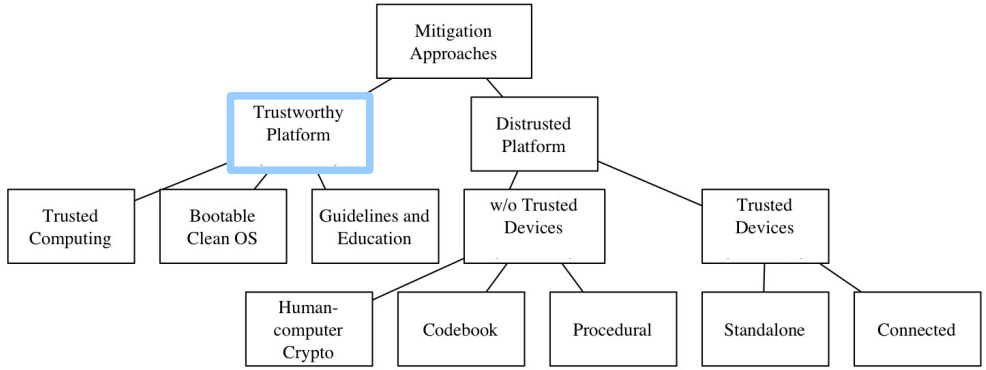


- Two main classes: ***making the platform trustworthy*** vs. ***distrusting the platform***
- I will introduce both but with focus on the latter

Overview

1. Motivation
2. Problem Description
3. Taxonomy of Mitigation Approaches
4. Making the Platform Trustworthy
5. Distrusting the Platform (Approaches w/o Additional Devices)
6. Distrusting the Platform (Personal Trusted Devices)
7. Conclusion and Future Work

Trustworthy Platform



Idea: the platform becomes ***T.***

Trusted Computing:
- Trusted Platform Module
- Special Security Software
- Effective Solution for the Secure Platform Problem
- No additional effort for the election authority
- Wide deployment in foreseeable future questionable

Secrecy
Anonymity
Integrity

Bootable Clean OS:
- Read-only media
- Cost-effective
- Effective against almost all attacks, but prone to BIOS corruption
- But: difficult to attack on a large-scale
- Usability questionable (configurations, drivers, settings, ...)
- Development, Secure Deployment, Maintenance, ...

Secrecy
Anonymity
Integrity

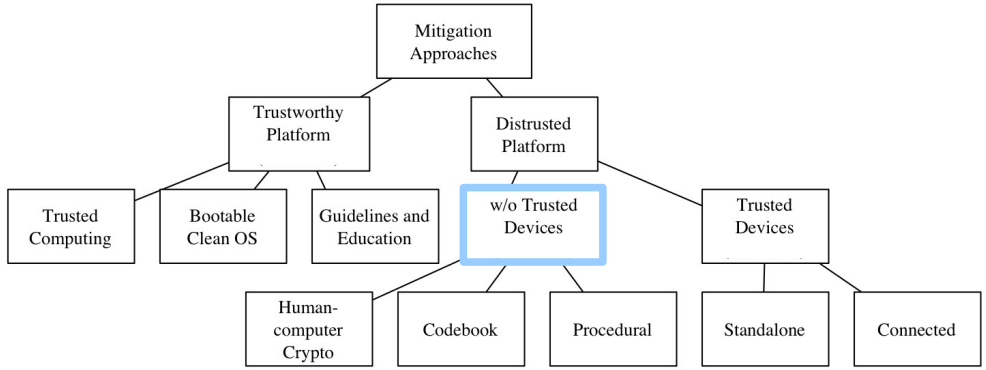
Guidelines and Education:
- Widely adapted approach
- Cost-effective
- Users do not or cannot follow instructions (user's laziness impacts not only herself!)
- Manipulated instructions

Secrecy
Anonymity
Integrity

Overview

1. Motivation
2. Problem Description
3. Taxonomy of Mitigation Approaches
4. Making the Platform Trustworthy
5. Distrusting the Platform (Approaches w/o Additional Devices)
6. Distrusting the Platform (Personal Trusted Devices)
7. Conclusion and Future Work

Distrusting the Platform



Approaches without additional devices.

Human-computer cryptography:
- The voter (human) performs cryptographic operations
- Machines cannot do more than humans (given enough Paper, Pencils, and time)
- But most humans are rather limited w.r.t memory and computing power
- Interesting ideas like Schneier's Solitaire algorithm
- Not user-friendly and proven to be not applicable for secure communication

Secrecy
Anonymity
Integrity

Codebook:
- Pre-encryption of messages and distribution of plain-text/cipher-text mapping (codebook)
- Easy look-up for encryption and decryption (where humans are good in)
- One example is Code Voting
- Cost-efficient
- Needs a secure "out-of-band" channel that is out of scope of $\mathcal{A}$ (e.g. mail)
- Integrity with confirmation codes possible

Secrecy
Anonymity
Integrity

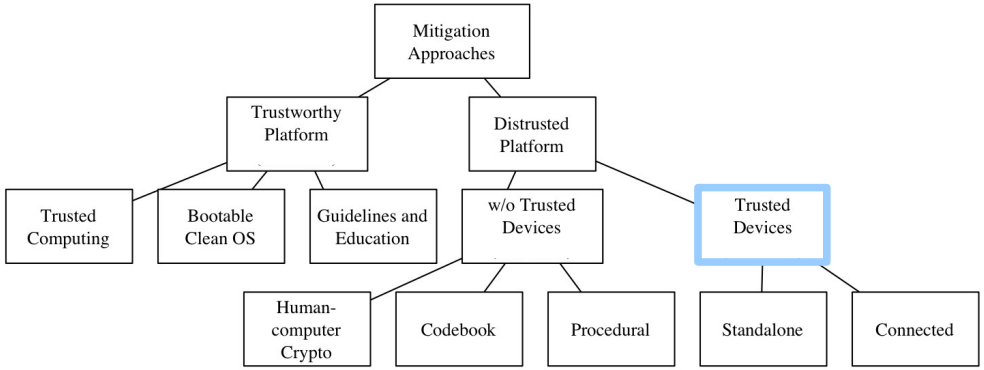
Procedural:
- Vote Updating (Makes sense only when $\mathcal{V}$ is able to detect manipulations)
- Anonymous Voting (e.g. using Blind Signature Schemes)
- Test Ballots (Kind of Intrusion Detection System)
- Manipulated instructions

Secrecy
Anonymity
Integrity

Overview

1. Motivation
2. Problem Description
3. Taxonomy of Mitigation Approaches
4. Making the Platform Trustworthy
5. Distrusting the Platform (Approaches w/o Additional Devices)
6. Distrusting the Platform (Personal Trusted Devices)
7. Conclusion and Future Work

Distrusting the Platform



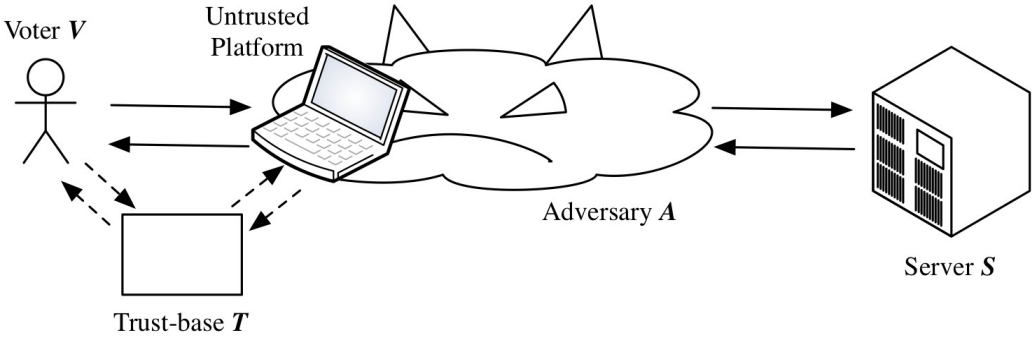
Trusted Devices.

Trusted Devices (TDs):
- Actually trustworthy devices
- Offer trustworthy execution of certain operations
- Universal

Personal Trusted Devices (PTDs):
- TDs that store user-specific information too (e.g. secret keys)
- Individual (personal)

Standalone vs. connected:
- TDs can be standalone as like pocket calculators
- TDs can also be connected (logically or physically) to A, usually to the untrusted platform
- Depending of the interface to V, possible security properties are limited

Distrusting the Platform



Approaches based on standalone Trusted Devices.

Communication to V:

- T must offer input and output interface to V

Secrecy
Anonymity
Integrity

Example: Challenge-Response Authentication in e-Banking:

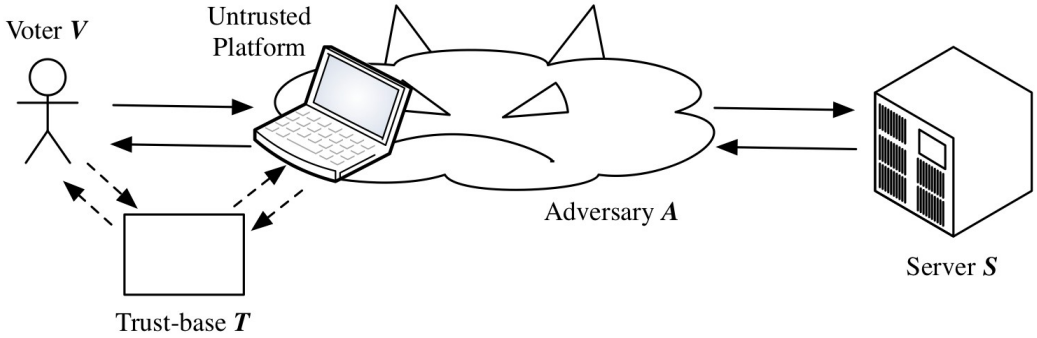
- S first sends a challenge code through V's untrusted platform to V
- V enters code into card reader together with bank card's PIN (personal identification number) to access the card
- Card computes a MAC (message authentication code) of challenge code and secret information that is stored on card and displays the MAC to V through the reader's display
- Finally, V enters MAC into untrusted platform and submits it to S
- If response code corresponds to a valid MAC, bank is convinced that correct card (containing the secret information) and a person knowing the correct PIN to access the card have been involved in the protocol run

Transaction authentication :

- Same procedure as before but transaction-critical information must be entered into T as well
- Could be adapted to e-voting, since voting is a kind of transaction
- Idea: MAC-Chain-Voting (Top Secret ;-))

Distrusting the Platform

Approaches based on connected Trusted Devices.



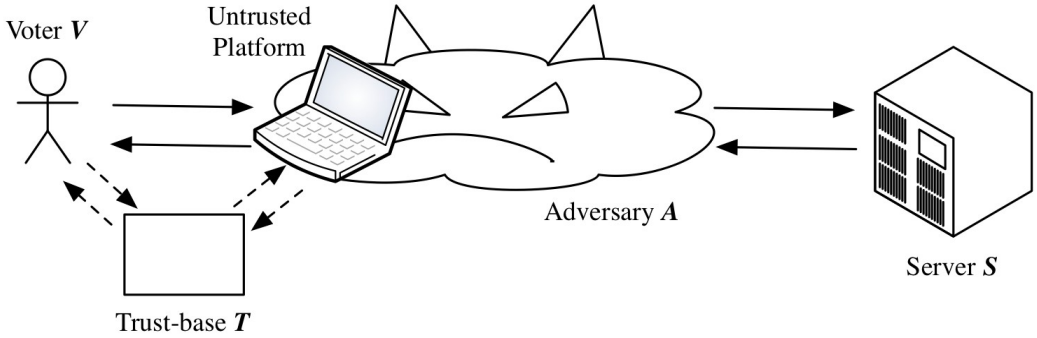
Connected Trusted Devices:

- T must not necessarily offer input and output interface to V
- T may communicate with A uni or bidirectional
- Remember that A includes V's platform and any other network service (e.g. GSM, Wi-Fi, ...)

Possible Implementations:

- No communication between T and V
- Unidirectional Communication from V to T
- Unidirectional Communication from T to V
- Bidirectional communication between T and V

Distrusting the Platform



No communication between T and V

Secrecy
Anonymity
Integrity

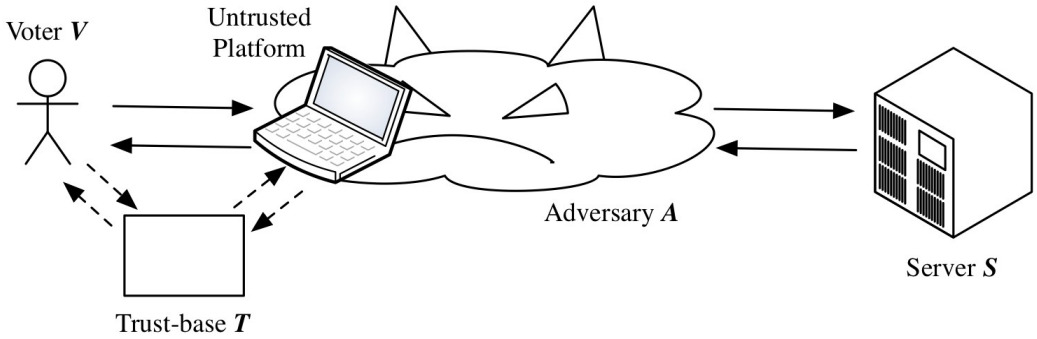
Description:

- T has no interface to communicate with V
- V communicates only with A
- Only guarantee is that T was connected to A during the protocol run

Example and observations:

- Smart card in combination with class-1 reader
- Every message sent by V gets immediately known by A
- A can pass arbitrary messages to the card
- After once entering PIN to access card, A can arbitrarily use the cards functionalities as long as the card is attached to A
- Widely applied in e-banking and e-business (MIGROSBANK, SuisseID, …)

Distrusting the Platform



Unidirectional Communication from V to T

Secrecy
Anonymity
Integrity

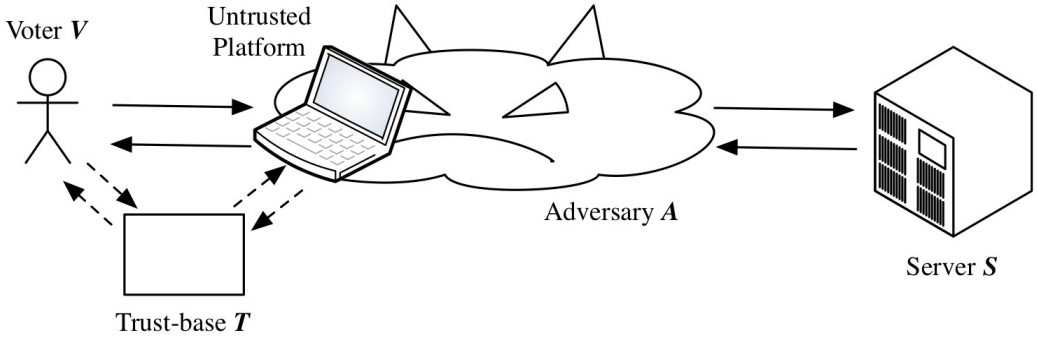
Description:

- T offers input interface to V (e.g. pinpad, microphone, ...)
- V may send clear-text messages directly to T
- T encrypts messages and sends them to A

Example and observations:

- Smart card in combination with class-2 reader
- PIN must be entered into T for every access to the card
- A cannot use the card's functionalities without V's consent
- Messages can be entered directly into T
- Integrity can be achieved with a trick
- User-friendliness questionable
- Expensive

Distrusting the Platform



Unidirectional Communication from T to V

Secrecy
Anonymity
Integrity

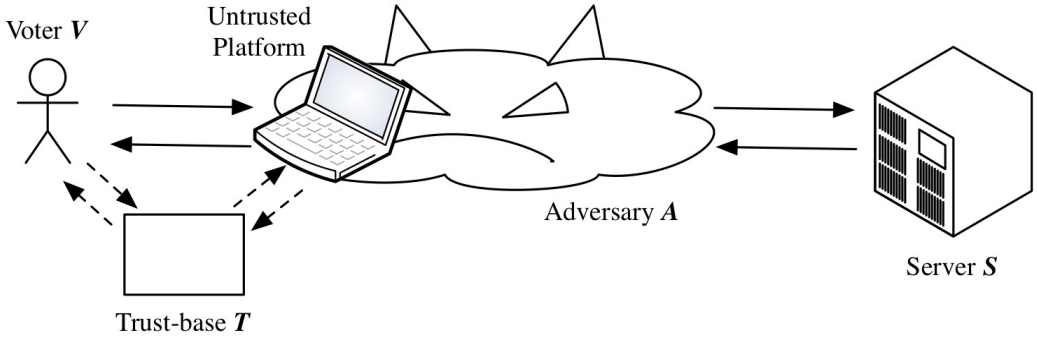
Description:

- T has an output interface such as a display or speaker to V
- V can only send messages to A and A again learns all messages
- A includes V's platform and any other network service (e.g. GSM, Wi-Fi, ...)

Example and observations:

- Smart card reader with only a display
- Display can be used to verify what actually gets processed by the card
- Another example is SMS-TAN
- Secrecy can again be achieved by a trick
- Sophisticated and unintuitive user interfaces
- Usability questionable
- Expensive

Distrusting the Platform



Bidirectional comm. between T and V

Secrecy
Anonymity
Integrity

Description:

- T offers input and output interface to V
- V may send messages directly to T
- V may conveniently verify what T performs

Example and observations:

- Smart card in combination with class-3 reader
- Another example is the networked solution of IBM (ZTIC)
- Intuitive for V
- Complex
- Very expensive

Overview

1. Motivation
2. Problem Description
3. Taxonomy of Mitigation Approaches
4. Making the Platform Trustworthy
5. Distrusting the Platform (Approaches w/o Additional Devices)
6. Distrusting the Platform (Personal Trusted Devices)
7. Conclusion and Future Work

Conclusion and Future Work

Promising approaches:

- Trusted Computing (availability unclear, can voters be forced to use TC?)
- Codebook (reasonable in most settings where elections take place every couple of years)
- Personal Trusted Devices (security depends on implementation, useful in direct democracy settings, cost-efficient if used for different purposes)

But: E-banking != e-voting

- Differing requirements (e.g. anonymity of the user)
- Some complementary measures of e-banking cannot be adapted

Open Issues:

- Combinations of approaches must be analyzed (e.g. MAC-Chain-Voting)
- Impact of approaches on server-side must be examined (voter privacy, verifiability)

Questions