A Security Flaw in the Verification Code Mechanism of the Norwegian Internet Voting System

Reto E. Koenig, Philipp Locher, Rolf Haenni

Bern University of Applied Sciences

08.09.2013

Supported by the Swiss National Science Foundation (project No. 200021L_140650)

Outline

1. Introduction
   - Problem Space
   - Properties of the Norwegian E-Voting Protocol
   - Implementation
2. Attack
   - Controlling the SMS-Channel: Network-Layer
   - Controlling the SMS-Channel: Application-Layer
   - Adversarial Communication
   - Adversarial Infection
   - Counter Meassurements
3. Conclusion

Outline

1. Introduction
   - Problem Space
   - Properties of the Norwegian E-Voting Protocol
   - Implementation
2. Attack
   - Controlling the SMS-Channel: Network-Layer
   - Controlling the SMS-Channel: Application-Layer
   - Adversarial Communication
   - Adversarial Infection
   - Counter Meassurements
3. Conclusion

Outline

1. Introduction
   - Problem Space
   - Properties of the Norwegian E-Voting Protocol
   - Implementation
2. Attack
   - Controlling the SMS-Channel: Network-Layer
   - Controlling the SMS-Channel: Application-Layer
   - Adversarial Communication
   - Adversarial Infection
   - Counter Meassurements
3. Conclusion

Why This Talk?

The Norwegian E-Voting System...

- ...allows vote updating
- ...uses SMS as out-of-band post channel
- ...allows smart-phone as trusted device
- ...faces a secure platform problem
- ...cannot provide the required vote integrity by verification code
- ...can be fixed!

Why This Talk?

The Norwegian E-Voting System...

- ...allows vote updating
- ...uses SMS as out-of-band post channel
- ...*allows* smart-phone as trusted device
- ...faces a secure platform problem
- ...cannot provide the required vote integrity by verification-code
- ...can be fixed!

Why This Talk?

The Norwegian E-Voting System...

- ...allows vote updating
- ...uses SMS as out-of-band post channel
- ...*allows* smart-phone as trusted device
- ...faces a secure platform problem
- ...cannot provide the required vote integrity by verification-code
- ...can be fixed!

Why This Talk?

The Norwegian E-Voting System...

- ...allows vote updating

- ...uses SMS as out-of-band post channel
- ...*allows* smart-phone as trusted device

- ...faces a secure platform problem
- ...cannot provide the required vote integrity by verification-code

- ...can be fixed!

Why This Talk?

The Norwegian E-Voting System...

- ...allows vote updating
- ...uses SMS as out-of-band post channel
- ...*allows* smart-phone as trusted device
- ...faces a secure platform problem
- ...cannot provide the required vote integrity by verification-code
- ...can be fixed!

Why This Talk?

The Norwegian E-Voting System...

- ...allows vote updating
- ...uses SMS as out-of-band post channel
- ...*allows* smart-phone as trusted device
- ...faces a secure platform problem
- ...cannot provide the required vote integrity by verification-code
- ...can be fixed!

Why This Talk?

The Norwegian E-Voting System...

- ...allows vote updating
- ...uses SMS as out-of-band post channel
- ...*allows* smart-phone as trusted device
- ...faces a secure platform problem
- ...cannot provide the required vote integrity by verification-code
- ...can be fixed!

Adversary Model and Trust Assumptions

The Norwegian E-Voting System...

- ...assumes server side to be honest
- ...accepts a malicious browser (MITB)
  - *cannot guarantee privacy (at the voter side)*

Adversary Model and Trust Assumptions

The Norwegian E-Voting System...

- ...assumes server side to be honest
- ...accepts a malicious browser (MITB)
- → *cannot guarantee privacy (at the voter side)*

Adversary Model and Trust Assumptions

The Norwegian E-Voting System…

- …assumes server side to be honest
- …accepts a malicious browser (MITB)
- → *cannot guarantee privacy (at the voter side)*

Adversary Model and Trust Assumptions

The Norwegian E-Voting System...

- ...assumes server side to be honest
- ...accepts a malicious browser (MITB)
- → *cannot guarantee privacy (at the voter side)*

Adversary Model and Trust Assumptions

The Norwegian E-Voting System...

- ...creates voter-individual and candidate based verification code (at the server side)
- ...sends verification codes to the voter via out-of-band channels
- ...provides vote integrity, vote and [illegible]

Adversary Model and Trust Assumptions

The Norwegian E-Voting System...

- ...creates voter-individual and candidate based verification code (at the server side)
- ...sends verification codes to the voter via out-of-band channels
  - pre-channel: postal mail (secure printing)
  - post-channel: SMS (cryptographic receipt generation)
- → provides vote integrity: *cast and recorded as intended*

Adversary Model and Trust Assumptions

The Norwegian E-Voting System...

- ...creates voter-individual and candidate based verification code (at the server side)
- ...sends verification codes to the voter via out-of-band channels

  pre-channel postal mail (secure printing)

  post-channel SMS (cryptographic receipt-generator)
- → provides vote integrity: *cast and recorded as intended*

Adversary Model and Trust Assumptions

The Norwegian E-Voting System...

- ...creates voter-individual and candidate based verification code (at the server side)
- ...sends verification codes to the voter via out-of-band channels

  pre-channel postal mail (secure printing)

  post-channel SMS (cryptographic receipt-generator)
- → provides vote integrity: *cast and recorded as intended*

Adversary Model and Trust Assumptions

The Norwegian E-Voting System...

- ...creates voter-individual and candidate based verification code (at the server side)
- ...sends verification codes to the voter via out-of-band channels

  pre-channel postal mail (secure printing)
  post-channel SMS (cryptographic receipt-generator)
- → provides vote integrity: *cast and recorded as intended*

Adversary Model and Trust Assumptions

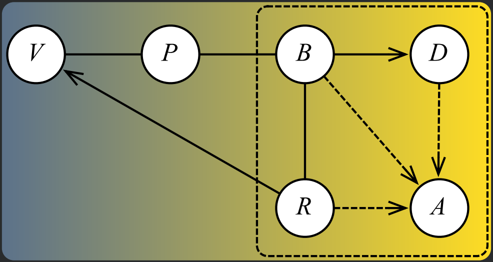
The Norwegian E-Voting System...

- ...creates voter-individual and candidate based verification code (at the server side)
- ...sends verification codes to the voter via out-of-band channels

  pre-channel postal mail (secure printing)
  post-channel SMS (cryptographic receipt-generator)
- → provides vote integrity: *cast and recorded as intended*

Adversary Model and Trust Assumptions

The Norwegian E-Voting System...



Verification Code: Recoginizing an Attack

If...

- ...correct receipt appears after the vote: Good
- ... wrong receipt appears after the vote: Bad
- ... no receipt appears after the vote: Bad
- ... receipt appears without prior vote: Bad

Verification Code: Recoginizing an Attack

If...

- ...correct receipt appears after the vote: Good
- ... wrong receipt appears after the vote: Bad
- ... no receipt appears after the vote: Bad
- ... receipt appears without prior vote: Bad

Verification Code: Recoginizing an Attack

If...

- ...correct receipt appears after the vote: Good
- ... wrong receipt appears after the vote: Bad
- ... no receipt appears after the vote: Bad
- ... receipt appears without prior vote: Bad

Verification Code: Recoginizing an Attack

If...

- ...correct receipt appears after the vote: Good
- ... wrong receipt appears after the vote: Bad
- ... no receipt appears after the vote: Bad
- ... receipt appears without prior vote: Bad

Verification Code: Recoginizing an Attack

If…

- …correct receipt appears after the vote: Good
- … wrong receipt appears after the vote: Bad
- … no receipt appears after the vote: Bad
- … receipt appears without prior vote: Bad

Speciality

The Norwegian E-Voting System...

...allows vote updating!

Out-of-Band Post-Channel

The Norwegian E-Voting System...

...uses SMS-channel

Enhanced Adversary model

Man in the...

...*B*rowser & *S*MS-Channel

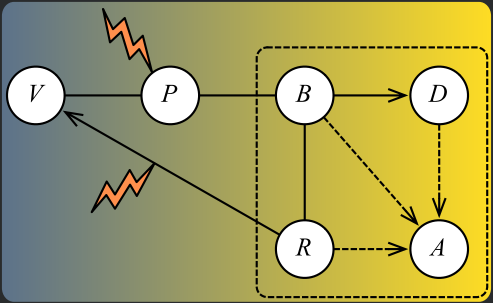
Enhanced Adversary model

Man in the...

...*B*rowser & *S*MS-Channel

SMS-Channel: No more Out-of-Band

MIT(B + S) + Vote-Updating $\mapsto$ Compromized System



Fake GSM Base Transceiver *Dedicated Hardware*

IMSI-catcher: Demonstrated live @ Defcon 2010

- Hardware: 1500$ (off-the-shelf HW)
- Software: 0$ (Open Source)
- Trick: Proxy GSM-network ↔ GSM-phone
- Range: ≈ 35km

Fake GSM Base Transceiver *Dedicated Hardware*

IMSI-catcher: Demonstrated live @ Defcon 2010

- Hardware: 1500$ (off-the-shelf HW)
- Software: 0$ (Open Source)
- Task: Proxy GSM-network ↔ GSM-phone
- Range ≈ 35km

Fake GSM Base Transceiver *Dedicated Hardware*

IMSI-catcher: Demonstrated live @ Defcon 2010

- Hardware: 1500$ (off-the-shelf HW)
- Software: 0$ (Open Source)

- Task: Proxy GSM-network ↔ GSM-phone
- Range ≈ 35km

Fake GSM Base Transceiver *Dedicated Hardware*

IMSI-catcher: Demonstrated live @ Defcon 2010

- Hardware: 1500$ (off-the-shelf HW)
- Software: 0$ (Open Source)
- Task: Proxy GSM-network ↔ GSM-phone
- Range ≈ 35km

Fake GSM Base Transceiver *Dedicated Hardware*

IMSI-catcher: Demonstrated live @ Defcon 2010

- Hardware: 1500$ (off-the-shelf HW)
- Software: 0$ (Open Source)

- Task: Proxy GSM-network ↔ GSM-phone
- Range ≈ 35km

Malicious Browser (MITB) & Fake GSM Base Transceiver (MITS) *Dedicated Hardware*

Logic

- MITB informs MITS to withold every second SMS:

  receipt generator $\mapsto$ voter's phone

1. Voter votes in an e-voting session (via MITB)
2. SMS: receipt generator $\mapsto$ voter's phone
3. Malicious vote update: MITB silently performs a vote-update in the voter's e-voting session (MITB votes)
4. SMS: receipt generator $\mapsto$ voter's phone (MITS blocks)

$\rightarrow$ *Silent Vote Update*

Malicious Browser (MITB) & Fake GSM Base Transceiver (MITS) *Dedicated Hardware*

Logic

- MITB informs MITS to withold every second SMS:

receipt generator ↦ voter's phone

- Voter votes in an e-voting session (via MITB)
- SMS: receipt generator ⟶ voter's phone
- Malicious vote update – MITB silently performs a vote in the voter's e-voting session (MITB works)
- SMS: receipt generator ⟶ voter's phone (MITS blocks)

→ *Silent Vote Update*

Malicious Browser (MITB) & Fake GSM Base Transceiver (MITS) *Dedicated Hardware*

Logic

- MITB informs MITS to withold every second SMS:

receipt generator ↦ voter's phone

1. Voter votes in an e-voting session (via MITB)
2. SMS: receipt generator ↦ voter's phone
3. Malicious vote update: MITB silently performs a vote-cast in the voter's e-voting session (MITB votes)
4. SMS: receipt generator $\not\mapsto$ voter's phone (MITS blocks)

→ *Silent Vote Update*

Malicious Browser (MITB) & Fake GSM Base Transceiver (MITS) *Dedicated Hardware*

Logic

- MITB informs MITS to withold every second SMS:

  receipt generator ↦ voter's phone

1. Voter votes in an e-voting session (via MITB)
2. SMS: receipt generator ↦ voter's phone
3. Malicious vote update: MITB silently performs a vote-cast in the voter's e-voting session (MITB votes)
4. SMS: receipt generator ↛ voter's phone (MITS blocks)

→ *Silent Vote Update*

Malicious Browser (MITB) & Fake GSM Base Transceiver (MITS) *Dedicated Hardware*

Logic

- MITB informs MITS to withold every second SMS:

  receipt generator ↦ voter's phone

1. Voter votes in an e-voting session (via MITB)
2. SMS: receipt generator ↦ voter's phone
3. Malicious vote update: MITB silently performs a vote-cast in the voter's e-voting session (MITB votes)
4. SMS: receipt generator ↛ voter's phone (MITS blocks)

→ *Silent Vote Update*

Malicious Browser (MITB) & Fake GSM Base Transceiver (MITS) *Dedicated Hardware*

Logic

- MITB informs MITS to withold every second SMS:

  receipt generator $\mapsto$ voter's phone

1. Voter votes in an e-voting session (via MITB)
2. SMS: receipt generator $\mapsto$ voter's phone
3. Malicious vote update: MITB silently performs a vote-cast in the voter's e-voting session (MITB votes)
4. SMS: receipt generator $\not\mapsto$ voter's phone (MITS blocks)

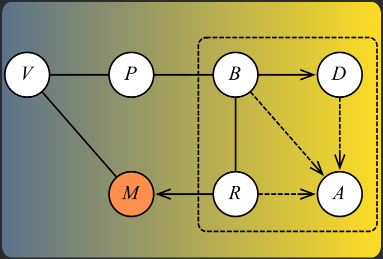
$\rightarrow$ *Silent Vote Update*

Statement

Introduction of smart-phone technology results in a new environment with strong impact on the security model

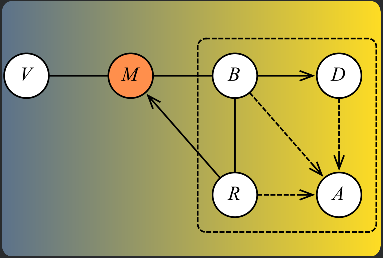
The Norwegian E-Voting System...

Introduction of Smart-Phone Technology to the Norwegian System



The Norwegian E-Voting System...

Introduction of Smart-Tablet Technology to the Norwegian System



Enhanced Adversary model

Man in the...

...*B*rowser & *SMS*-App

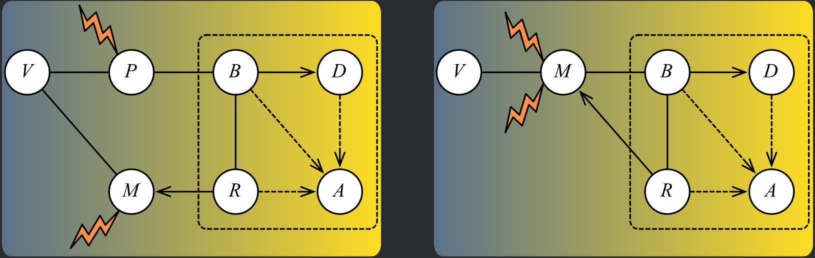
Enhanced Adversary model

Man in the…

…*B*rowser & *S*MS-App

SMS-Channel: No more Out-of-Band

MIT(B + S) + Vote-Updating ↦ Compromized System



Malicious Browser & SMS-App

1

**Web-Buddy (MITB) & SMS-Buddy (MITS)
Demonstrated live @ SIC [1] 2013**

- Hardware: 0$ (Voter's device)
- Software: 0$ (Open Source)
- Task: Proxy smart-phone → voter
- Range: [illegible]

[1] Swiss Informatics Competition

Malicious Browser & SMS-App

1

Web-Buddy (MITB) & SMS-Buddy (MITS)
Demonstrated live @ SIC [1] 2013

- Hardware: 0$ (Voter's device)
- Software: 0$ (Open Source)
- Task: Proxy smart-phone ↔ voter
- Range $\approx \infty$

[1] Swiss Informatics Competition

Malicious Browser & SMS-App

1

Web-Buddy (MITB) & SMS-Buddy (MITS) Demonstrated live @ SIC [1] 2013

- Hardware: 0$ (Voter's device)
- Software: 0$ (Open Source)
- Task: Proxy smart-phone ↔ voter
- Range $\approx \infty$

[1] Swiss Informatics Competition

Malicious Browser & SMS-App

1

Web-Buddy (MITB) & SMS-Buddy (MITS) Demonstrated live @ SIC [1] 2013

- Hardware: 0$ (Voter's device)
- Software: 0$ (Open Source)

- Task: Proxy smart-phone ↔ voter
- Range $\approx \infty$

[1] Swiss Informatics Competition

Malicious Browser & SMS-App

1

Web-Buddy (MITB) & SMS-Buddy (MITS)
Demonstrated live @ SIC [1] 2013

- Hardware: 0$ (Voter's device)
- Software: 0$ (Open Source)

- Task: Proxy smart-phone ↔ voter
- Range $\approx \infty$

[1] Swiss Informatics Competition

Malicious Browser & SMS-App

Logic

- MITB informs MITS to withold every second SMS from receipt generator

1. Voter votes in an e-voting session (via MITB)
2. SMS: receipt generator → voter
3. Malicious vote update: MITB silently performs a vote-update in the voter's e-voting session (MITB votes)
4. SMS: receipt generator → voter (MITS blocks)

→ *Silent Vote Update*

Malicious Browser & SMS-App

Logic

- MITB informs MITS to withold every second SMS from receipt generator
- Voter votes in an e-voting session (via MITB)
- SMS: receipt generator → voter
- Malicious vote update: MITB silently performs a vote in the voter's e-voting session (MITB vote)
- SMS: receipt generator → voter (MITS blocks)

→ *Silent Vote Update*

Malicious Browser & SMS-App

Logic

- MITB informs MITS to withold every second SMS from receipt generator

1. Voter votes in an e-voting session (via MITB)
2. SMS: receipt generator ↦ voter
3. Malicious vote update: MITB silently performs a vote-cast in the voter's e-voting session (MITB votes)
4. SMS: receipt generator ↛ voter (MITS blocks)

→ *Silent Vote Update*

Malicious Browser & SMS-App

Logic

- MITB informs MITS to withold every second SMS from receipt generator

1. Voter votes in an e-voting session (via MITB)
2. SMS: receipt generator $\mapsto$ voter
3. Malicious vote update: MITB silently performs a vote-cast in the voter's e-voting session (MITB votes)
4. SMS: receipt generator $\not\mapsto$ voter (MITS blocks)

$\rightarrow$ *Silent Vote Update*

Malicious Browser & SMS-App

Logic

- MITB informs MITS to withold every second SMS from receipt generator

1. Voter votes in an e-voting session (via MITB)
2. SMS: receipt generator $\mapsto$ voter
3. Malicious vote update: MITB silently performs a vote-cast in the voter's e-voting session (MITB votes)
4. SMS: receipt generator $\not\mapsto$ voter (MITS blocks)

→ *Silent Vote Update*

Malicious Browser & SMS-App

Logic

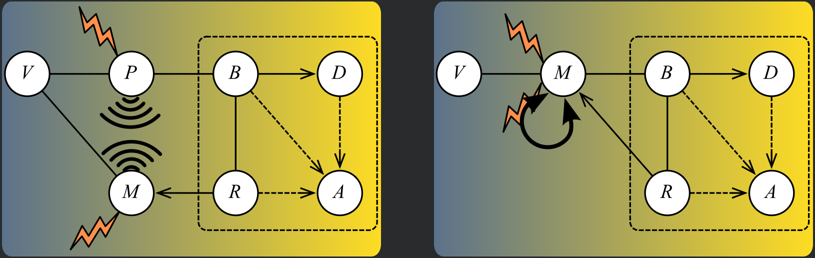
- MITB informs MITS to withold every second SMS from receipt generator

1. Voter votes in an e-voting session (via MITB)
2. SMS: receipt generator $\mapsto$ voter
3. Malicious vote update: MITB silently performs a vote-cast in the voter's e-voting session (MITB votes)
4. SMS: receipt generator $\nrightarrow$ voter (MITS blocks)

$\rightarrow$ *Silent Vote Update*

'Silent-Channel' communication

How They Communicate



Channel: SMS-App ↔ Browser

How They Communicate

None SMS-App is programmed to block every second SMS from receipt generator

Implementation available

Channel: SMS-App ↔ Browser

How They Communicate

None SMS-App is programmed to block every second SMS from receipt generator

Implementation available

Channel: SMS-App ↔ Browser

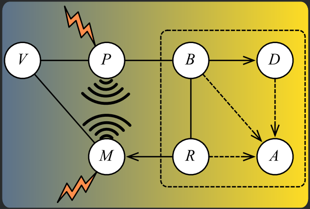
How They Communicate

None SMS-App is programmed to block every second SMS from receipt generator

Implementation available

Channel: SMS-App ↔ Browser

How They Communicate: Two Devices (Smart-phone, Tablet)



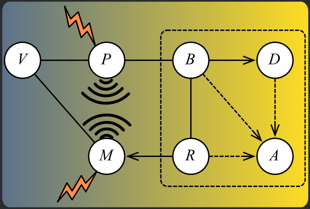
Internet-SMS-Gateway: MITB only allowed to use the internet

SMS-Internet-Gateway: MITS only allowed to read/write SMS

Implementation available

Channel: SMS-App ↔ Browser

How They Communicate: Two Devices (Smart-phone, Tablet)



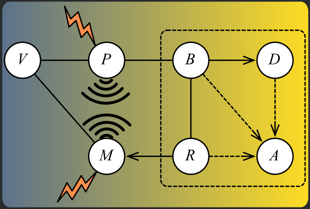
Internet-SMS-Gateway MITB only allowed to use the internet

SMS-Internet-Gateway MITS only allowed to read/write SMS

Implementation available

Channel: SMS-App ↔ Browser

How They Communicate: Two Devices (Smart-phone, Tablet)

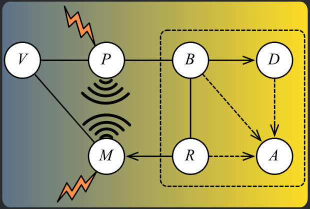


Internet-SMS-Gateway	MITB only allowed to use the internet
SMS-Internet-Gateway	MITS only allowed to read/write SMS

Implementation available

Channel: SMS-App ↔ Browser

How They Communicate: Two Devices (Smart-phone, Tablet)



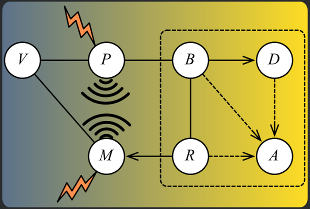
Internet-SMS-Gateway MITB only allowed to use the internet

SMS-Internet-Gateway MITS only allowed to read/write SMS

Implementation available

Channel: SMS-App ↔ Browser

How They Communicate: Two Devices (Smart-phone, Tablet)



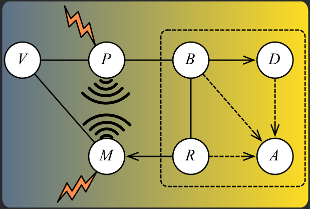
Internet-SMS-Gateway MITB only allowed to use the internet

SMS-Internet-Gateway MITS only allowed to read/write SMS

Implementation available

Channel: SMS-App → Browser

How They Communicate: Two Devices (Smart-Phone, Notebook)

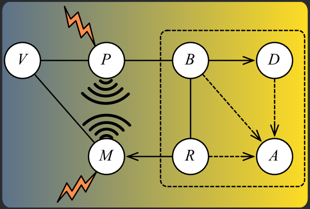


Ultra-Sonic MITS broadcasts via loudspeaker
MITB listens via microphone

Proof of Concept available

Channel: SMS-App → Browser

How They Communicate: Two Devices (Smart-Phone, Notebook)

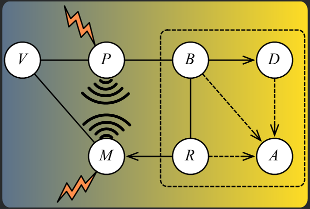


Ultra-Sonic MITS broadcasts via loudspeaker
MITB listens via microphone

Proof of Concept available

Channel: SMS-App → Browser

How They Communicate: Two Devices (Smart-Phone, Notebook)

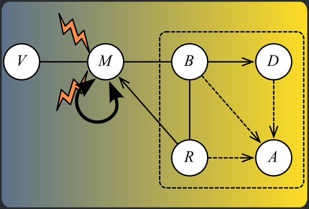


Ultra-Sonic MITS broadcasts via loudspeaker
MITB listens via microphone

Proof of Concept available

Channel: SMS-App ↔ Browser

How they communicate: Single device (tablet)

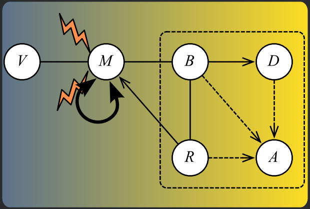


Inter-Process-Communication MITS & MITB on same device (Tablet)

Implementation available (SMS-Buddy, Web-Buddy)

Channel: SMS-App ↔ Browser

How they communicate: Single device (tablet)

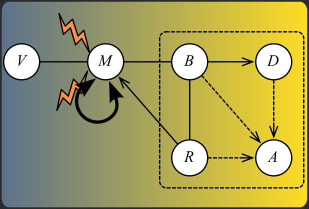


Inter-Process-Communication MITS & MITB on same device (Tablet)

Implementation available (SMS-Buddy, Web-Buddy)

Channel: SMS-App ↔ Browser

How they communicate: Single device (tablet)



Inter-Process-Communication MITS & MITB on same device (Tablet)

Implementation available (SMS-Buddy, Web-Buddy)

How to Infect the Smart-Phone

Assumption: The browser is already infected

Social Engineering SMS-App is designed to guard privacy! No additional permissions needed!
Browser keeps 'nagging'

Cloud-Based After user has logged into cloud (Google-Play-Store / Apple-App-Store / ...)
Browser directly downloads SMS-App

Implementations and success-stories available (Eurograbber et al)

How to Infect the Smart-Phone

Assumption: The browser is already infected

Social-Engineering SMS-App is designed to guard privacy! No additional permissions needed!
Browser keeps 'nagging'

Cloud-Based After user has logged into cloud (Google-Play-Store / Apple-App-Store / ...)
Browser directly downloads SMS-App

Implementations and success-stories available (Eurograbber et al)

How to Infect the Smart-Phone

Assumption: The browser is already infected

Social-Engineering SMS-App is designed to guard privacy! No additional permissions needed!
Browser keeps 'nagging'

Cloud-Based After user has logged into cloud (Google-Play-Store / Apple-App-Store / ...)
Browser directly downloads SMS-App

Implementations and success-stories available (Eurograbber et al)

How to Infect the Smart-Phone

Assumption: The browser is already infected

Social-Engineering SMS-App is designed to guard privacy! No additional permissions needed!
Browser keeps 'nagging'

Cloud-Based After user has logged into cloud (Google-Play-Store / Apple-App-Store / ...)
Browser directly downloads SMS-App

Implementations and success-stories available (Eurograbber et al)

How to Infect the Smart-Phone

Assumption: The browser is already infected

Social-Engineering SMS-App is designed to guard privacy! No additional permissions needed!
Browser keeps 'nagging'

Cloud-Based After user has logged into cloud (Google-Play-Store / Apple-App-Store / ...)
Browser directly downloads SMS-App

Implementations and success-stories available (Eurograbber et al)

How to Void this Attack on the Norwegian E-Voting System

One Vote only...

Per Voting Session MinID used to authenticate and authorize voter. Equal to e-banking mTAN.
This is no real solution, as Web-Buddy and SMS-Buddy are designed to break e-banking mTAN → attacking MinID

[illegible] No vote updating → no successful attack

How to Void this Attack on the Norwegian E-Voting System

One Vote only...

Per Voting-Session: MinID used to authenticate and authorize voter. Equal to e-banking mTAN.
This is no real solution, as Web-Buddy and SMS-Buddy are designed to break e-banking mTAN ↦ attacking MinID

!: No vote updating ... no successful attack.

How to Void this Attack on the Norwegian E-Voting System

One Vote only...

Per Voting-Session — MinID used to authenticate and authorize voter. Equal to e-banking mTAN.
This is no real solution, as Web-Buddy and SMS-Buddy are designed to break e-banking mTAN $\mapsto$ attacking MinID

! — No vote updating ... no successful attack.

How to Void this Attack on the Norwegian E-Voting System

Dedicated Hardware Device

SMS-Receiver: A must, if MinID alike infrastructure shall remain.
However, Fake GSM-Attack still possible.

Trusted Hardware Token: Secure Display, Secure Keyboard. Messages E2E encrypted (over the Internet).

Implementation available: ZTIC(IBM UBS)
Cronto-Device (Steven Murdoch)

Success-Story available: E-Banking (UBS)

How to Void this Attack on the Norwegian E-Voting System

Dedicated Hardware Device

SMS-Receiver: A must, if MinID alike infrastructure shall remain.
However, Fake GSM-Attack still possible.

Trusted Hardware Token: Secure Display, Secure Keyboard
Messages E2E encrypted (over the Internet).

Implementation available: ZTIC(IBM UBS)
Cronto-Device (Steven Murdoch)

Success-Story available: E-Banking (UBS)

How to Void this Attack on the Norwegian E-Voting System

Dedicated Hardware Device

SMS-Receiver A must, if MinID alike infrastructure shall remain.
However, Fake GSM-Attack still possible.

Trusted Hardware Token Secure Display, Secure Keyboard
Messages E2E encrypted (over the Internet).

Implementation available: ZTIC(IBM UBS)
Cronto-Device (Steven Murdoch)

Success-Story available: E-Banking (UBS)

How to Void this Attack on the Norwegian E-Voting System

Dedicated Hardware Device

SMS-Receiver A must, if MinID alike infrastructure shall remain.
However, Fake GSM-Attack still possible.

Trusted Hardware Token Secure Display, Secure Keyboard
Messages E2E encrypted (over the Internet).

Implementation available: ZTIC(IBM UBS)
Cronto-Device (Steven Murdoch)

Success-Story available: E-Banking (UBS)

How to Void this Attack on the Norwegian E-Voting System

Dedicated Hardware Device

SMS-Receiver	A must, if MinID alike infrastructure shall remain. However, Fake GSM-Attack still possible.
Trusted Hardware Token	Secure Display, Secure Keyboard Messages E2E encrypted (over the Internet).

Implementation available: ZTIC(IBM UBS)
Cronto-Device (Steven Murdoch)
Success-Story available: E-Banking (UBS)

Smart-Phones Do Not Provide any Out-of-Band Channel

Stop using smart-phones as trusted device

Your system will be grounded by a *script kiddie*

Smart-Phones Do Not Provide any Out-of-Band Channel

Stop using smart-phones as trusted device

Your system will be grounded by a *script kiddie*