SURF NET

Radboud University Nijmegen

Using Trusted Execution Environments in Two-Factor Authentication

Roland van Rijswijk-Deij M.Sc.

rijswijk@cs.ru.nl

http://www.cs.ru.nl/~rijswijk/

Institute for Computing and Information Sciences – Digital Security
Radboud University Nijmegen

SURFnet bv

Open Identity Summit 2013, Kloster Banz, Germany
September 10th 2013

What is a Trusted Execution Environment?

- **Isolated Execution**
  applications execute isolated from and unhindered by others; application code and data is protected at run-time
- **Secure Storage**
  all persistently stored data (e.g. cryptographic keys) of an application is protected against access by other applications
- **Remote Attestation**
  enables remote parties to ascertain they are dealing with a particular trusted application on a particular TEE
- **Secure Provisioning**
  remote parties can communicate with a specific application on a specific TEE while protecting integrity and confidentiality
- **Trusted Path**
  protected I/O channel(s) for data input by the user and data output to the user

based on definition from Vasudevan et al., LNCS 7344, pp 159-178, Springer 2012 [VOZ+]

What is a Trusted Execution Environment?

- **Isolated Execution**
  applications execute isolated from and unhindered by others; application code and data is protected at run-time
- **Secure Storage**
  all persistently stored data (e.g. cryptographic keys) of an application is protected against access by other applications
- **Remote Attestation**
  enables remote parties to ascertain they are dealing with a particular trusted application on a particular TEE
- **Secure Provisioning**
  remote parties can communicate with a specific application on a specific TEE while protecting integrity and confidentiality
- **Trusted Path**
  protected I/O channel(s) for data input by the user and data output to the user

based on definition from Vasudevan et al., LNCS 7344, pp 159-178, Springer 2012 [VOZ+]

What is a Trusted Execution Environment?

- **Isolated Execution**
  applications execute isolated from and unhindered by others; application code and data is protected at run-time
- **Secure Storage**
  all persistently stored data (e.g. cryptographic keys) of an application is protected against access by other applications
- **Remote Attestation**
  enables remote parties to ascertain they are dealing with a particular trusted application on a particular TEE
- **Secure Provisioning**
  remote parties can communicate with a specific application on a specific TEE while protecting integrity and confidentiality
- **Trusted Path**
  protected I/O channel(s) for data input by the user and data output to the user

based on definition from Vasudevan et al., LNCS 7344, pp 159-178, Springer 2012 [VOZ+]

What is a Trusted Execution Environment?

- **Isolated Execution**
  applications execute isolated from and unhindered by others; application code and data is protected at run-time
- **Secure Storage**
  all persistently stored data (e.g. cryptographic keys) of an application is protected against access by other applications
- **Remote Attestation**
  enables remote parties to ascertain they are dealing with a particular trusted application on a particular TEE
- **Secure Provisioning**
  remote parties can communicate with a specific application on a specific TEE while protecting integrity and confidentiality
- **Trusted Path**
  protected I/O channel(s) for data input by the user and data output to the user

based on definition from Vasudevan et al., LNCS 7344, pp 159-178, Springer 2012 [VOZ+]

What is a Trusted Execution Environment?

- **Isolated Execution**
  applications execute isolated from and unhindered by others; application code and data is protected at run-time
- **Secure Storage**
  all persistently stored data (e.g. cryptographic keys) of an application is protected against access by other applications
- **Remote Attestation**
  enables remote parties to ascertain they are dealing with a particular trusted application on a particular TEE
- **Secure Provisioning**
  remote parties can communicate with a specific application on a specific TEE while protecting integrity and confidentiality
- **Trusted Path**
  protected I/O channel(s) for data input by the user and data output to the user

based on definition from Vasudevan et al., LNCS 7344, pp 159-178, Springer 2012 [VOZ+]

What is a Trusted Execution Environment?

- **Isolated Execution**
  applications execute isolated from and unhindered by others; application code and data is protected at run-time
- **Secure Storage**
  all persistently stored data (e.g. cryptographic keys) of an application is protected against access by other applications
- **Remote Attestation**
  enables remote parties to ascertain they are dealing with a particular trusted application on a particular TEE
- **Secure Provisioning**
  remote parties can communicate with a specific application on a specific TEE while protecting integrity and confidentiality
- **Trusted Path**
  protected I/O channel(s) for data input by the user and data output to the user

based on definition from Vasudevan et al., LNCS 7344, pp 159-178, Springer 2012 [VOZ+]

Our interest in TEEs

- Our research group is working on a privacy-friendly attribute-based identity card
- One of the open problems is how to convey to the user what information is disclosed about them
- Smart cards don't have a UI :-(
- `https://www.irmacard.org/`

Relation to Two-Factor Authentication

- Most two-factor authentication solutions are based on *something you have* and *something you know*
- In practice, something you have is often a physical token
- In the age of laptops and mobile phones this is often perceived as inconvenient
- Why not use *something you* **already** *have*?
- Vendors have started to embrace this with a whole host of apps*
- But can these *really* be trusted?
- Platform manufacturers provide solutions for this, let's look at two of them

*e.g. van Rijswijk & van Dijk, tiqr: a novel take on two-factor authentication, USENIX, 2011 [vRvD11]

Relation to Two-Factor Authentication

- Most two-factor authentication solutions are based on *something you have* and *something you know*
- In practice, something you have is often a physical token
- In the age of laptops and mobile phones this is often perceived as inconvenient
- Why not use *something you* **already** *have*?
- Vendors have started to embrace this with a whole host of apps*
- But can these *really* be trusted?
- Platform manufacturers provide solutions for this, let's look at two of them

*e.g. van Rijswijk & van Dijk, tiqr: a novel take on two-factor authentication, USENIX, 2011 [vRvD11]

Relation to Two-Factor Authentication

- Most two-factor authentication solutions are based on *something you have* and *something you know*
- In practice, something you have is often a physical token
- In the age of laptops and mobile phones this is often perceived as inconvenient
- Why not use *something you* **already** *have*?
- Vendors have started to embrace this with a whole host of apps*
- But can these *really* be trusted?
- Platform manufacturers provide solutions for this, let's look at two of them

*e.g. van Rijswijk & van Dijk, tiqr: a novel take on two-factor authentication, USENIX, 2011 [vRvD11]

Relation to Two-Factor Authentication

- Most two-factor authentication solutions are based on *something you have* and *something you know*
- In practice, something you have is often a physical token
- In the age of laptops and mobile phones this is often perceived as inconvenient
- Why not use *something you* **already** *have*?
- Vendors have started to embrace this with a whole host of apps*
- But can these *really* be trusted?
- Platform manufacturers provide solutions for this, let's look at two of them

*e.g. van Rijswijk & van Dijk, tiqr: a novel take on two-factor authentication, USENIX, 2011 [vRvD11]

Relation to Two-Factor Authentication

- Most two-factor authentication solutions are based on *something you have* and *something you know*
- In practice, something you have is often a physical token
- In the age of laptops and mobile phones this is often perceived as inconvenient
- Why not use *something you* **already** *have*?
- Vendors have started to embrace this with a whole host of apps*
- But can these *really* be trusted?
- Platform manufacturers provide solutions for this, let's look at two of them

*e.g. van Rijswijk & van Dijk, tiqr: a novel take on two-factor authentication, USENIX, 2011 [vRvD11]

Relation to Two-Factor Authentication

- Most two-factor authentication solutions are based on *something you have* and *something you know*
- In practice, something you have is often a physical token
- In the age of laptops and mobile phones this is often perceived as inconvenient
- Why not use *something you* **already** *have*?
- Vendors have started to embrace this with a whole host of apps*
- But can these *really* be trusted?
- Platform manufacturers provide solutions for this, let's look at two of them

*e.g. van Rijswijk & van Dijk, tiqr: a novel take on two-factor authentication, USENIX, 2011 [vRvD11]

Relation to Two-Factor Authentication

- Most two-factor authentication solutions are based on *something you have* and *something you know*
- In practice, something you have is often a physical token
- In the age of laptops and mobile phones this is often perceived as inconvenient
- Why not use *something you* **already** *have*?
- Vendors have started to embrace this with a whole host of apps*
- But can these *really* be trusted?
- Platform manufacturers provide solutions for this, let's look at two of them

*e.g. van Rijswijk & van Dijk, tiqr: a novel take on two-factor authentication, USENIX, 2011 [vRvD11]

Relation to Two-Factor Authentication

- Most two-factor authentication solutions are based on *something you have* and *something you know*
- In practice, something you have is often a physical token
- In the age of laptops and mobile phones this is often perceived as inconvenient
- Why not use *something you* **already** *have*?
- Vendors have started to embrace this with a whole host of apps*
- But can these *really* be trusted?
- Platform manufacturers provide solutions for this, let's look at two of them

*e.g. van Rijswijk & van Dijk, tiqr: a novel take on two-factor authentication, USENIX, 2011 [vRvD11]

Intel IPT

- Built-in to certain Intel chipsets ("vPro" in "Sandy Bridge", "Ivy Bridge", "Haswell")
- RISC CPU separate from the main x86 cores
- Platform for trusted applications
- Runs applications developed by Intel partners (e.g. Vasco, Symantec, . . .)
- Closed development environment subject to NDAs and licensing
- Current applications: built-in OTP & PKI token, "Protected Transaction Display"
- Not to be confused with "Trusted Execution Technology" (TXT)!
- **Note:** technical information about IPT is very hard to find!

Intel IPT

- Built-in to certain Intel chipsets ("vPro" in "Sandy Bridge", "Ivy Bridge", "Haswell")
- RISC CPU separate from the main x86 cores
- Platform for trusted applications
- Runs applications developed by Intel partners (e.g. Vasco, Symantec, . . .)
- Closed development environment subject to NDAs and licensing
- Current applications: built-in OTP & PKI token, "Protected Transaction Display"
- Not to be confused with "Trusted Execution Technology" (TXT)!
- **Note:** technical information about IPT is very hard to find!

Intel IPT

- Built-in to certain Intel chipsets ("vPro" in "Sandy Bridge", "Ivy Bridge", "Haswell")
- RISC CPU separate from the main x86 cores
- Platform for trusted applications
- Runs applications developed by Intel partners (e.g. Vasco, Symantec, . . .)
- Closed development environment subject to NDAs and licensing
- Current applications: built-in OTP & PKI token, "Protected Transaction Display"
- Not to be confused with "Trusted Execution Technology" (TXT)!
- **Note:** technical information about IPT is very hard to find!

Intel IPT

- Built-in to certain Intel chipsets ("vPro" in "Sandy Bridge", "Ivy Bridge", "Haswell")
- RISC CPU separate from the main x86 cores
- Platform for trusted applications
- Runs applications developed by Intel partners (e.g. Vasco, Symantec, . . .)
- Closed development environment subject to NDAs and licensing
- Current applications: built-in OTP & PKI token, "Protected Transaction Display"
- Not to be confused with "Trusted Execution Technology" (TXT)!
- **Note:** technical information about IPT is very hard to find!

Intel IPT

- Built-in to certain Intel chipsets ("vPro" in "Sandy Bridge", "Ivy Bridge", "Haswell")
- RISC CPU separate from the main x86 cores
- Platform for trusted applications
- Runs applications developed by Intel partners (e.g. Vasco, Symantec, ...)
- Closed development environment subject to NDAs and licensing
- Current applications: built-in OTP & PKI token, "Protected Transaction Display"
- Not to be confused with "Trusted Execution Technology" (TXT)!
- **Note:** technical information about IPT is very hard to find!

Intel IPT

- Built-in to certain Intel chipsets ("vPro" in "Sandy Bridge", "Ivy Bridge", "Haswell")
- RISC CPU separate from the main x86 cores
- Platform for trusted applications
- Runs applications developed by Intel partners (e.g. Vasco, Symantec, ...)
- Closed development environment subject to NDAs and licensing
- Current applications: built-in OTP & PKI token, "Protected Transaction Display"
- Not to be confused with "Trusted Execution Technology" (TXT)!
- **Note:** technical information about IPT is very hard to find!

Intel IPT

- Built-in to certain Intel chipsets ("vPro" in "Sandy Bridge", "Ivy Bridge", "Haswell")
- RISC CPU separate from the main x86 cores
- Platform for trusted applications
- Runs applications developed by Intel partners (e.g. Vasco, Symantec, ...)
- Closed development environment subject to NDAs and licensing
- Current applications: built-in OTP & PKI token, "Protected Transaction Display"
- Not to be confused with "Trusted Execution Technology" (TXT)!
- **Note:** technical information about IPT is very hard to find!

Intel IPT

- Built-in to certain Intel chipsets ("vPro" in "Sandy Bridge", "Ivy Bridge", "Haswell")
- RISC CPU separate from the main x86 cores
- Platform for trusted applications
- Runs applications developed by Intel partners (e.g. Vasco, Symantec, ...)
- Closed development environment subject to NDAs and licensing
- Current applications: built-in OTP & PKI token, "Protected Transaction Display"
- Not to be confused with "Trusted Execution Technology" (TXT)!
- **Note:** technical information about IPT is very hard to find!

Intel IPT

- Built-in to certain Intel chipsets ("vPro" in "Sandy Bridge", "Ivy Bridge", "Haswell")
- RISC CPU separate from the main x86 cores
- Platform for trusted applications
- Runs applications developed by Intel partners (e.g. Vasco, Symantec, ...)
- Closed development environment subject to NDAs and licensing
- Current applications: built-in OTP & PKI token, "Protected Transaction Display"
- Not to be confused with "Trusted Execution Technology" (TXT)!
- **Note:** technical information about IPT is very hard to find!

ARM TrustZone

- ARM is not a silicon manufacturer; they sell chip designs
- TrustZone is like a box of Lego for building TEEs
- Chief components:
  - CPU with separate security worlds (normal & secure)
  - System bus (for interaction with on-die components)
  - Peripheral bus (for interaction with external components like a touch screen)
- Some pieces of the puzzle are missing (i.e. not supplied by ARM), e.g. secure storage
- The overall security of a TZ system depends on both the system-on-a-chip design as well as the software running on it

ARM TrustZone

- ARM is not a silicon manufacturer; they sell chip designs
- TrustZone is like a box of Lego for building TEEs
- Chief components:
  - CPU with separate security worlds (normal & secure)
  - System bus (for interaction with on-die components)
  - Peripheral bus (for interaction with external components like a touch screen)
- Some pieces of the puzzle are missing (i.e. not supplied by ARM), e.g. secure storage
- The overall security of a TZ system depends on both the system-on-a-chip design as well as the software running on it

ARM TrustZone

- ARM is not a silicon manufacturer; they sell chip designs
- TrustZone is like a box of Lego for building TEEs
- Chief components:
  - CPU with separate security worlds (normal & secure)
  - System bus (for interaction with on-die components)
  - Peripheral bus (for interaction with external components like a touch screen)
- Some pieces of the puzzle are missing (i.e. not supplied by ARM), e.g. secure storage
- The overall security of a TZ system depends on both the system-on-a-chip design as well as the software running on it

ARM TrustZone

- ARM is not a silicon manufacturer; they sell chip designs
- TrustZone is like a box of Lego for building TEEs
- Chief components:
  - CPU with separate security worlds (normal & secure)
  - System bus (for interaction with on-die components)
  - Peripheral bus (for interaction with external components like a touch screen)
- Some pieces of the puzzle are missing (i.e. not supplied by ARM), e.g. secure storage
- The overall security of a TZ system depends on both the system-on-a-chip design as well as the software running on it

ARM TrustZone

- ARM is not a silicon manufacturer; they sell chip designs
- TrustZone is like a box of Lego for building TEEs
- Chief components:
  - CPU with separate security worlds (normal & secure)
  - System bus (for interaction with on-die components)
  - Peripheral bus (for interaction with external components like a touch screen)
- Some pieces of the puzzle are missing (i.e. not supplied by ARM), e.g. secure storage
- The overall security of a TZ system depends on both the system-on-a-chip design as well as the software running on it

ARM TrustZone

- ARM is not a silicon manufacturer; they sell chip designs
- TrustZone is like a box of Lego for building TEEs
- Chief components:
  - CPU with separate security worlds (normal & secure)
  - System bus (for interaction with on-die components)
  - Peripheral bus (for interaction with external components like a touch screen)
- Some pieces of the puzzle are missing (i.e. not supplied by ARM), e.g. secure storage
- The overall security of a TZ system depends on both the system-on-a-chip design as well as the software running on it

User Interaction

- Can a user trust that:
  - what they enter on a token is not intercepted?
  - transaction information that is displayed has not been altered?
- Increasingly important if we start merging what used to be separate tokens into devices the user already has

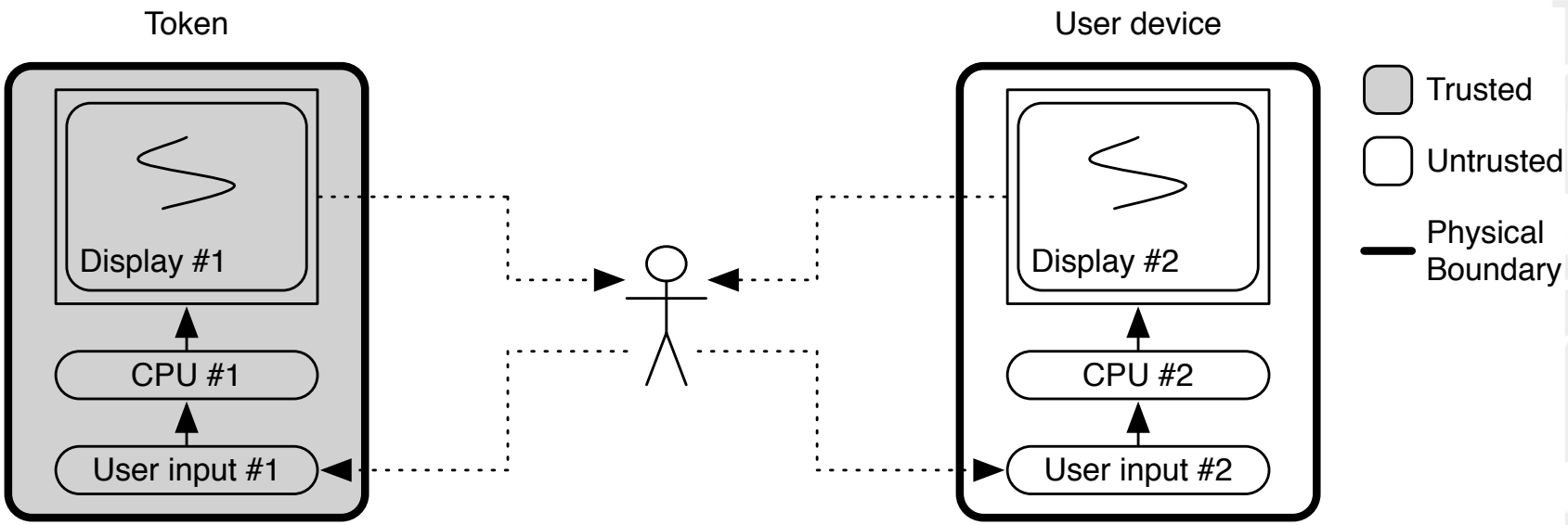
User Interaction

- Can a user trust that:
  - what they enter on a token is not intercepted?
  - transaction information that is displayed has not been altered?
- Increasingly important if we start merging what used to be separate tokens into devices the user already has

User Interaction

- Can a user trust that:
  - what they enter on a token is not intercepted?
  - transaction information that is displayed has not been altered?
- Increasingly important if we start merging what used to be separate tokens into devices the user already has

"Classic" Two-Factor Authentication

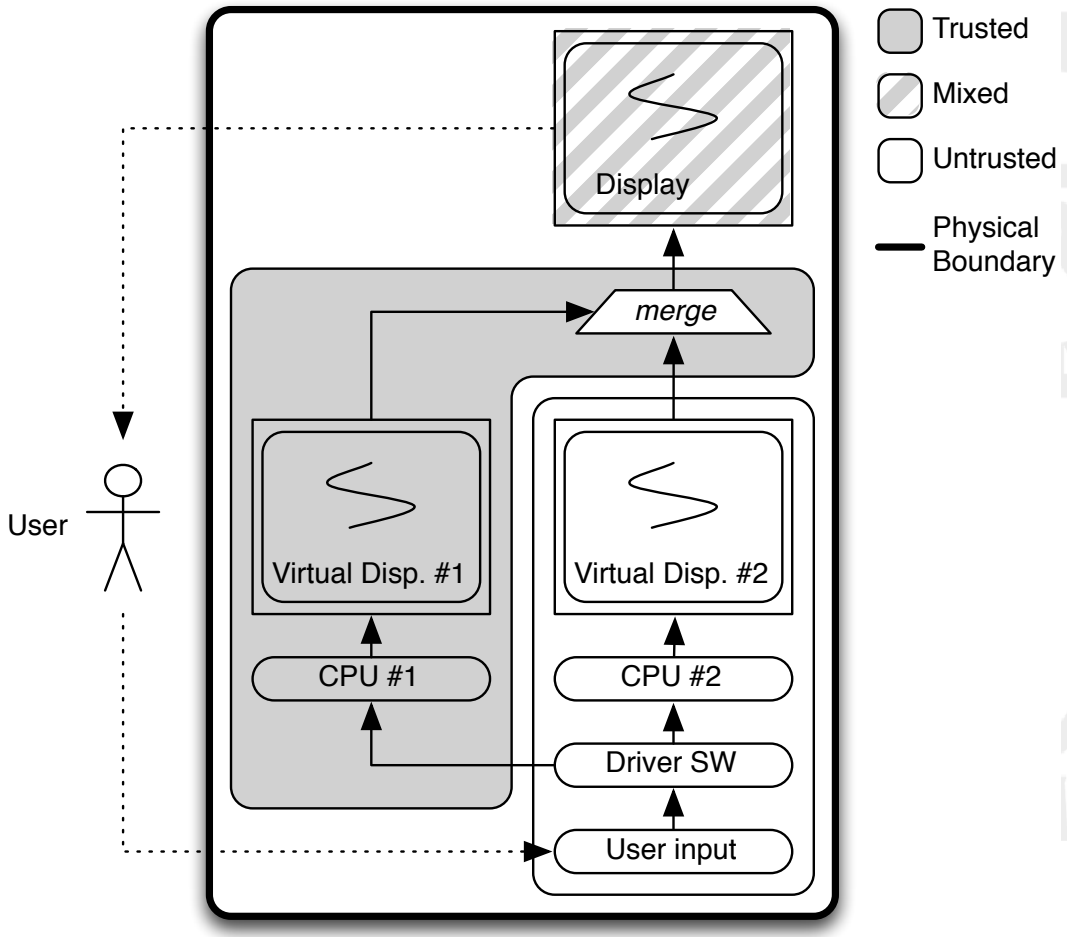


source: Vasco

- Strict physical separation
- It is always clear whether the user is dealing with a trusted or an untrusted environment

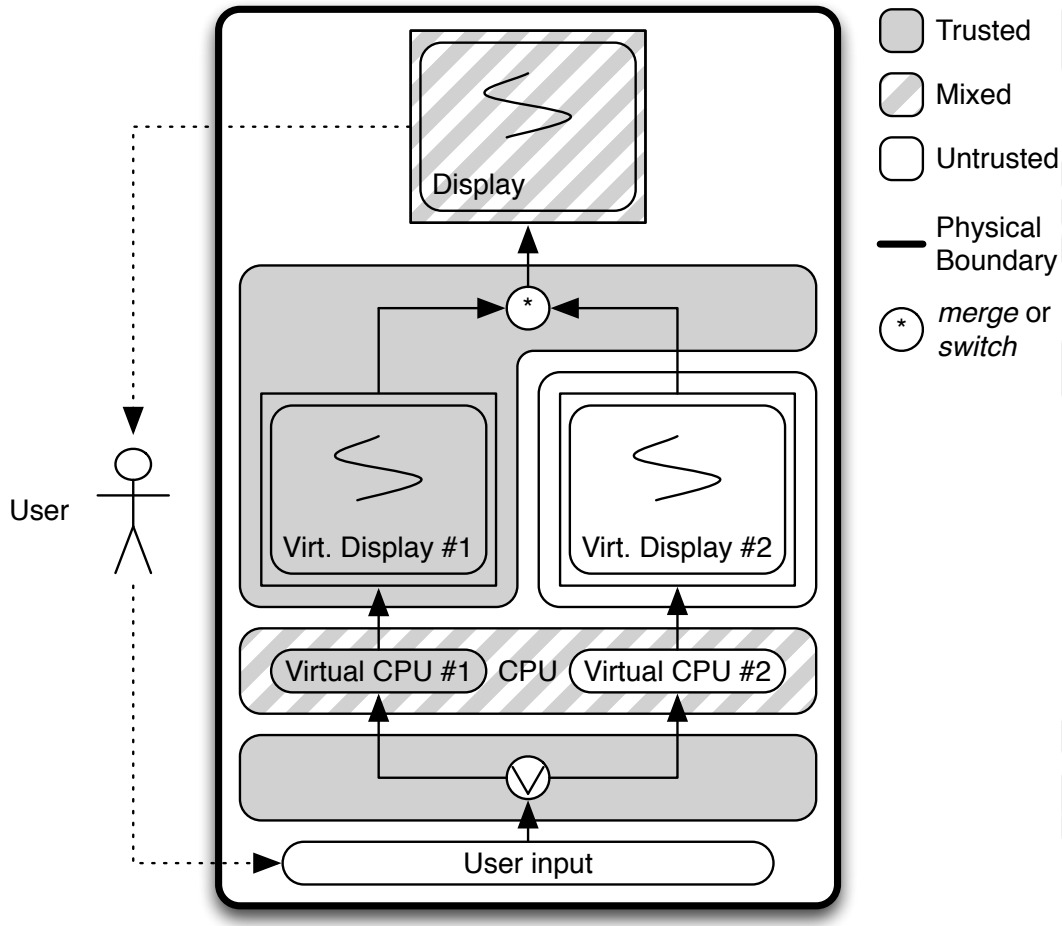
Intel IPT

- There is no trusted input path (!)
- Display is protected but probably only if the built-in graphics capabilities of the chipset are used

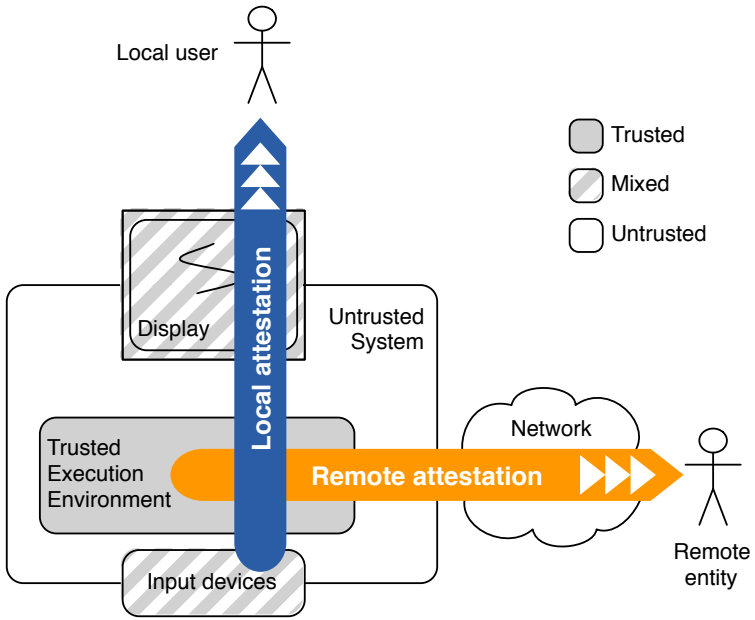


ARM TrustZone

- Figure is based on "ideal" chip design
- On a small(-ish) device the display could exclusively show either trusted or untrusted content



Local Attestation



- Both systems have a similar issue: how can a user tell they're interacting with a trusted application?
- All TEEs have this issue and there are many solutions
- This is going to be important if these kind of systems gain real traction

Industry View on Trusted Execution

- **Industry perception seems to be that TEEs can be used to deal with all these "untrustworthy" users**
  - TEEs are marketed as a means to enforce Digital Rights Management (DRM)
- **Developer access to TEEs is highly restricted**
  - Consequently, open development for industry-provided TEEs is almost impossible
- In our opinion this is a missed opportunity!
- TEEs can be an asset for both the user as well as relying parties when applied to authentication
- They can offer convenience to the user and can save relying parties that now issue physical tokens money

Industry View on Trusted Execution

- **Industry perception seems to be that TEEs can be used to deal with all these "untrustworthy" users**
  - TEEs are marketed as a means to enforce Digital Rights Management (DRM)
- **Developer access to TEEs is highly restricted**
  - Consequently, open development for industry-provided TEEs is almost impossible
- In our opinion this is a missed opportunity!
- TEEs can be an asset for both the user as well as relying parties when applied to authentication
- They can offer convenience to the user and can save relying parties that now issue physical tokens money

Industry View on Trusted Execution

- **Industry perception seems to be that TEEs can be used to deal with all these "untrustworthy" users**
  - TEEs are marketed as a means to enforce Digital Rights Management (DRM)
- **Developer access to TEEs is highly restricted**
  - Consequently, open development for industry-provided TEEs is almost impossible
- In our opinion this is a missed opportunity!
- TEEs can be an asset for both the user as well as relying parties when applied to authentication
- They can offer convenience to the user and can save relying parties that now issue physical tokens money

Industry View on Trusted Execution

- **Industry perception seems to be that TEEs can be used to deal with all these "untrustworthy" users**
  - TEEs are marketed as a means to enforce Digital Rights Management (DRM)
- **Developer access to TEEs is highly restricted**
  - Consequently, open development for industry-provided TEEs is almost impossible
- In our opinion this is a missed opportunity!
- TEEs can be an asset for both the user as well as relying parties when applied to authentication
- They can offer convenience to the user and can save relying parties that now issue physical tokens money

Industry View on Trusted Execution

- **Industry perception seems to be that TEEs can be used to deal with all these "untrustworthy" users**
  - TEEs are marketed as a means to enforce Digital Rights Management (DRM)
- **Developer access to TEEs is highly restricted**
  - Consequently, open development for industry-provided TEEs is almost impossible
- In our opinion this is a missed opportunity!
- TEEs can be an asset for both the user as well as relying parties when applied to authentication
- They can offer convenience to the user and can save relying parties that now issue physical tokens money

Industry View on Trusted Execution

- **Industry perception seems to be that TEEs can be used to deal with all these "untrustworthy" users**
  - TEEs are marketed as a means to enforce Digital Rights Management (DRM)
- **Developer access to TEEs is highly restricted**
  - Consequently, open development for industry-provided TEEs is almost impossible
- In our opinion this is a missed opportunity!
- TEEs can be an asset for both the user as well as relying parties when applied to authentication
- They can offer convenience to the user and can save relying parties that now issue physical tokens money

Conclusions

- Classic two-factor authentication solutions have favourable security properties
- The authentication industry is embracing "use-your-own-device"
- TEEs can approach this level of security but cannot match it (yet)
- The biggest problem is convincing the user they are dealing with a trusted environment

SURF NET

Radboud University Nijmegen

Conclusions

- Classic two-factor authentication solutions have favourable security properties
- The authentication industry is embracing "use-your-own-device"
- TEEs can approach this level of security but cannot match it (yet)
- The biggest problem is convincing the user they are dealing with a trusted environment

Conclusions

- Classic two-factor authentication solutions have favourable security properties
- The authentication industry is embracing "use-your-own-device"
- TEEs can approach this level of security but cannot match it (yet)
- The biggest problem is convincing the user they are dealing with a trusted environment

Conclusions

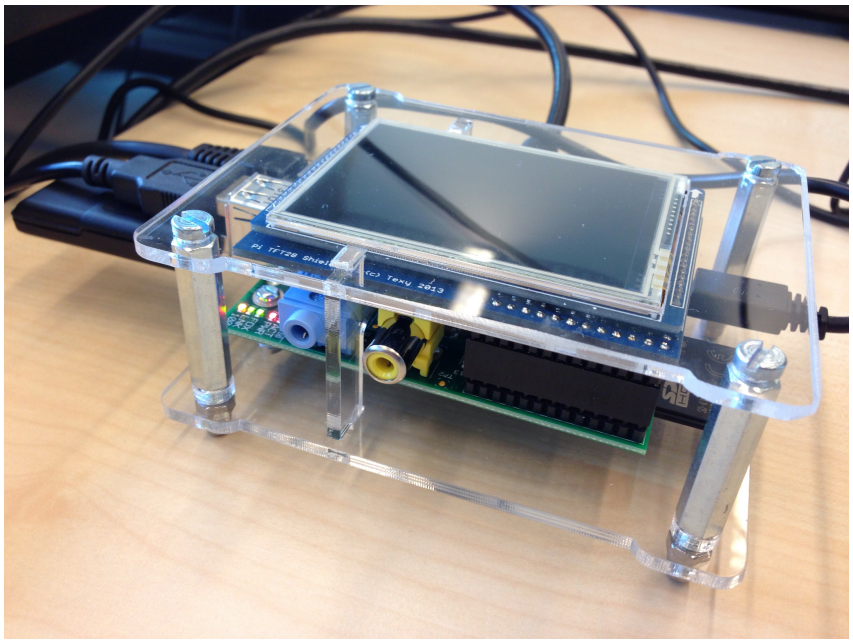
- Classic two-factor authentication solutions have favourable security properties
- The authentication industry is embracing "use-your-own-device"
- TEEs can approach this level of security but cannot match it (yet)
- The biggest problem is convincing the user they are dealing with a trusted environment

Conclusions

- Classic two-factor authentication solutions have favourable security properties
- The authentication industry is embracing "use-your-own-device"
- TEEs can approach this level of security but cannot match it (yet)
- The biggest problem is convincing the user they are dealing with a trusted environment

Future Work

The "π-vacy":



Consists of:

- Raspberry Pi mini-computer
- TFT touch-panel riser board
- NFC dongle that can act as target

Basically: **a very fat smart card**

Allows us to experiment with:

- Trusted user interaction
- TEEs based on μ-kernels
- Local attestation