Unpicking PLAID
A Cryptographic Analysis of an ISO-standards-track Authentication Protocol

Summer School on Real-World Crypto 2015

Jean Paul Degabriele
Kenneth G. Paterson

Victoria Fehr
Marc Fischlin
Tommaso Gagliardoni
Felix Günther
Giorgia Azzurra Marson
Arno Mittelbach

Information Security Group, Royal Holloway, University of London

Cryptoplexity, TU Darmstadt

Outline of this Talk

Introduction

Description of PLAID

Keyset Fingerprinting

Tracing Cards

General Security Concerns

Protocol for **L**ightweight **A**uthentication of **ID**entity

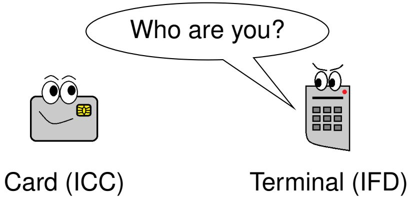
Protocol for **L**ightweight **A**uthentication of **ID**entity

- contactless authentication protocol

Card (ICC)

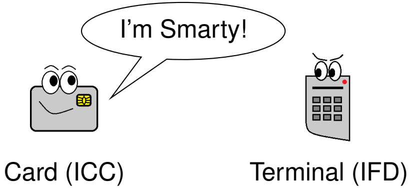
Terminal (IFD)

Protocol for **L**ightweight **A**uthentication of **ID**entity



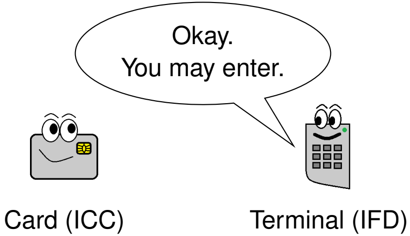
- contactless authentication protocol

Protocol for **L**ightweight **A**uthentication of **ID**entity



- contactless authentication protocol

Protocol for **L**ightweight **A**uthentication of **ID**entity



- contactless authentication protocol

TECHNISCHE UNIVERSITÄT DARMSTADT

Protocol for **L**ightweight **A**uthentication of **ID**entity

- contactless authentication protocol
- developed by Centrelink

Centrelink
giving you options

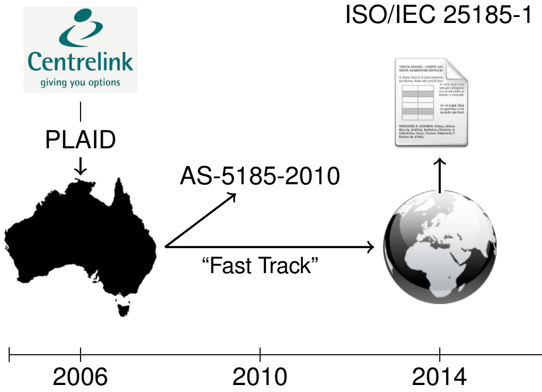
PLAID

2006

Protocol for **L**ightweight **A**uthentication of **ID**entity

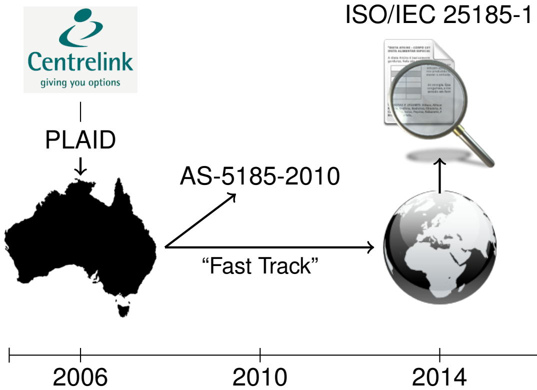
- contactless authentication protocol
- developed by Centrelink
- AS 5185-2010

Protocol for **L**ightweight **A**uthentication of **ID**entity



- contactless authentication protocol
- developed by Centrelink
- AS 5185-2010
- submitted to ISO via fast track as ISO/IEC 25185-1

Protocol for **L**ightweight **A**uthentication of **ID**entity



- contactless authentication protocol
- developed by Centrelink
- AS 5185-2010
- submitted to ISO via fast track as ISO/IEC 25185-1

The PLAID Protocol

- Building blocks: 2048-bit **RSA** with PKCS#1 v1.5 padding, **AES-128** in CBC mode and **SHA-256**.
- A keyset is a triple comprising of a 2-byte Keyset ID, an RSA key (encryption or decryption) and an AES key.
- A keyset corresponds to a **capability** (a token providing access to some object(s)).
- Keysets are preloaded in cards and terminals during initialisation.

The PLAID Protocol

- For each keyset there corresponds an AES **master key** K_i which is given to the terminals (IFDs).

- For a specific keyset each card will be assigned a different AES key and a unique card identifier called **Diversification Data** (DivData).

The PLAID Protocol

- For each keyset there corresponds an AES **master key** K_i which is given to the terminals (IFDs).

- For a specific keyset each card will be assigned a different AES key and a unique card identifier called **Diversification Data** (DivData).

- A terminal can derive a card's AES key K_i^{DD} from the master key and DivData, $K_i^{DD} = AES_{K_i}(\text{DivData})$.

- Each card is additionally preloaded with an extra set of **Shillkeys**, the use of which will be explained later.

The PLAID Protocol

ICC

IFD

The PLAID Protocol

ICC

IFD

index	RSA	AES
7	sk_7	K_7
34	sk_{34}	K_{34}
	⋮	

The PLAID Protocol

ICC

IFD

index	RSA	AES
2	pk_2	K_2^{DD}
7	pk_7	K_7^{DD}
	$\vdots$	

index	RSA	AES
7	sk_7	K_7
34	sk_{34}	K_{34}
	$\vdots$	

TECHNISCHE
UNIVERSITÄT
DARMSTADT

The PLAID Protocol

ICC

$\xleftarrow{(KeySetIDs)}$

IFD

index	RSA	AES
2	pk_2	K_2^{DD}
7	pk_7	K_7^{DD}
	$\vdots$	

index	RSA	AES
7	sk_7	K_7
34	sk_{34}	K_{34}
	$\vdots$	

The PLAID Protocol

ICC

$(34, 7, \ldots)$ ←

IFD

index	RSA	AES
2	pk_2	K_2^{DD}
7	pk_7	K_7^{DD}
	⋮	

index	RSA	AES
7	sk_7	K_7
34	sk_{34}	K_{34}
	⋮	

The PLAID Protocol

ICC

$$\xleftarrow{(34, 7, \ldots)}$$

$$\xrightarrow{RSA_{pk_7}(7, \text{DivData}, \text{RND1}, \text{RND1})}$$

IFD

ICC:

index	RSA	AES
2	pk_2	K_2^{DD}
7	pk_7	K_7^{DD}
	⋮	

IFD:

index	RSA	AES
7	sk_7	K_7
34	sk_{34}	K_{34}
	⋮	

The PLAID Protocol

ICC

$\xleftarrow{(34,\ 7,\ \ldots\ )}$

IFD

$\xrightarrow{\mathrm{RSA}_{pk_7}(7,\ \mathrm{DivData},\ \mathrm{RND1},\ \mathrm{RND1})}$

index	RSA	AES
2	pk_2	K_2^{DD}
7	pk_7	K_7^{DD}
	⋮	

index	RSA	AES
7	sk_7	K_7
34	sk_{34}	K_{34}
	⋮	

The PLAID Protocol

ICC

IFD

$\longleftarrow$ (34, 7, ...)

RSA_{pk_7}(7, DivData, RND1, RND1) $\longrightarrow$

index	RSA	AES
2	pk_2	K_2^{DD}
7	pk_7	K_7^{DD}
	⋮	

index	RSA	AES
7	sk_7	K_7
34	sk_{34}	K_{34}
	⋮	

The PLAID Protocol

ICC

$(34, 7, \ldots)$ ←

$\mathrm{RSA}_{pk_7}(7, \mathrm{DivData}, \mathrm{RND1}, \mathrm{RND1})$ →

IFD

index	RSA	AES
2	pk_2	K_2^{DD}
7	pk_7	K_7^{DD}
	⋮	

index	RSA	AES
7	sk_7	K_7
34	sk_{34}	K_{34}
	⋮	

$$K_7^{DD} = \mathrm{AES}_{K_7}(\mathrm{DivData})$$

$$k_{\mathrm{session}} = \mathrm{SHA}(RND1 \,||\, RND2)$$

The PLAID Protocol

ICC

IFD

(34, 7, ...) ←

RSA_{pk_7}(7, DivData, RND1, RND1) →

$\mathrm{AES}_{K_7^{DD}}$(AuthReq, RND2, payload, k_{session}) ←

index	RSA	AES
2	pk_2	K_2^{DD}
7	pk_7	K_7^{DD}
	⋮	

index	RSA	AES
7	sk_7	K_7
34	sk_{34}	K_{34}
	⋮	

$$K_7^{DD} = \mathrm{AES}_{K_7}(\mathrm{DivData})$$

$$k_{\mathrm{session}} = \mathrm{SHA}(RND1 \,||\, RND2)$$

The PLAID Protocol

ICC

IFD

$\xleftarrow{(34, 7, \ldots)}$

$\xrightarrow{\mathrm{RSA}_{pk_7}(7,\ \mathrm{DivData},\ \mathrm{RND1},\ \mathrm{RND1})}$

$\xleftarrow{\mathrm{AES}_{K_7^{DD}}(\mathrm{AuthReq},\ \mathrm{RND2},\ \mathrm{payload},\ k_{\mathrm{session}})}$

ICC:

index	RSA	AES
2	pk_2	K_2^{DD}
7	pk_7	K_7^{DD}
	⋮	

IFD:

index	RSA	AES
7	sk_7	K_7
34	sk_{34}	K_{34}
	⋮	

$K_7^{DD} = \mathrm{AES}_{K_7}(\mathrm{DivData})$

$k_{\mathrm{session}} = \mathrm{SHA}(RND1 || RND2)$

$k_{\mathrm{session}} = \mathrm{SHA}(RND1 || RND2)$

The PLAID Protocol

ICC

index	RSA	AES
2	pk_2	K_2^{DD}
7	pk_7	K_7^{DD}
	$\vdots$	

$(34, 7, \ldots)$ ←

$\mathrm{RSA}_{pk_7}(7, \mathrm{DivData}, \mathrm{RND1}, \mathrm{RND1})$ →

$\mathrm{AES}_{K_7^{DD}}(\mathrm{AuthReq}, \mathrm{RND2}, \mathrm{payload}, k_{\mathrm{session}})$ ←

$\mathrm{AES}_{k_{\mathrm{session}}}(\mathrm{AuthResp}, \mathrm{payload}, \mathrm{DivData})$ →

$k_{\mathrm{session}} = \mathrm{SHA}(RND1 || RND2)$

IFD

index	RSA	AES
7	sk_7	K_7
34	sk_{34}	K_{34}
	$\vdots$	

$K_7^{DD} = \mathrm{AES}_{K_7}(\mathrm{DivData})$

$k_{\mathrm{session}} = \mathrm{SHA}(RND1 || RND2)$

The PLAID Protocol

ICC

IFD

index	RSA	AES
2	pk_2	K_2^{DD}
7	pk_7	K_7^{DD}
	⋮	

$(34, 7, \ldots)$ ←

$\mathrm{RSA}_{pk_7}(7, \mathrm{DivData}, \mathrm{RND1}, \mathrm{RND1})$ →

$\mathrm{AES}_{K_7^{DD}}(\mathrm{AuthReq}, \mathrm{RND2}, \mathrm{payload}, k_{\mathrm{session}})$ ←

$\mathrm{AES}_{k_{\mathrm{session}}}(\mathrm{AuthResp}, \mathrm{payload}, \mathrm{DivData})$ →

index	RSA	AES
7	sk_7	K_7
34	sk_{34}	K_{34}
	⋮	

$K_7^{DD} = \mathrm{AES}_{K_7}(\mathrm{DivData})$

ICC: $k_{\mathrm{session}} = \mathrm{SHA}(RND1 || RND2)$

IFD: $k_{\mathrm{session}} = \mathrm{SHA}(RND1 || RND2)$

Channel secured with $k_{session}$ (optional)

The Security of PLAID

TECHNISCHE UNIVERSITÄT DARMSTADT

*“PLAID [...] is **cryptographically stronger**, faster and **more private** [...]”*

Centrelink PLAID Specification v8.0, 2009

The Security of PLAID

*"PLAID [...] is **cryptographically stronger**, faster and **more private** [...]"*

Centrelink PLAID Specification v8.0, 2009

*"[...] **strong authentication** [...] in a fast, **highly secure and private** fashion without the exposure of [...] identifying information or any other information which is useful to an attacker."*

ISO/IEC 25185-1.2, 2014

The Security of PLAID

*“PLAID [...] is **cryptographically stronger**, faster and **more private** [...]”*

Centrelink PLAID Specification v8.0, 2009

*“[...] **strong authentication** [...] in a fast, **highly secure and private** fashion without the exposure of [...] identifying information or any other information which is useful to an attacker.”*

ISO/IEC 25185-1.2, 2014

But no formal security analysis is provided!

Notions of Privacy

Notions of Privacy

Anonymity

Images by: *Giorgia Azzurra Marson*

Notions of Privacy

Anonymity

- Protocol does not reveal personal identification data of cardholders

Images by: *Giorgia Azzurra Marson*

Notions of Privacy

Anonymity

Untraceability

- Protocol does not reveal personal identification data of cardholders

Images by: *Giorgia Azzurra Marson*

Notions of Privacy

Anonymity

- Protocol does not reveal personal identification data of cardholders

Untraceability

- It should not be possible to trace the card's activity.

Images by: *Giorgia Azzurra Marson*

When Access is Denied...

ICC

index	RSA	AES
2	pk_2	K_2^{DD}
7	pk_7	K_7^{DD}

IFD

index	RSA	AES
5	sk_5	K_5
34	sk_{34}	K_{34}

When Access is Denied...

ICC

index	RSA	AES
2	pk_2	K_2^{DD}
7	pk_7	K_7^{DD}

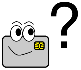
$\xleftarrow{\text{KeySetID} = (34, 5)}$

IFD

index	RSA	AES
5	sk_5	K_5
34	sk_{34}	K_{34}

- What if none of the presented keysets are supported by the card?

When Access is Denied...



ICC

index	RSA	AES
2	pk_2	K_2^{DD}
7	pk_7	K_7^{DD}

$\xleftarrow{\text{KeySetID} = (34, 5)}$

IFD

index	RSA	AES
5	sk_5	K_5
34	sk_{34}	K_{34}

- What if none of the presented keysets are supported by the card?

When Access is Denied...

ICC

index	RSA	AES
2	pk_2	K_2^{DD}
7	pk_7	K_7^{DD}
$*$	pk^*	K^*

KeySetID = (34, 5)

IFD

index	RSA	AES
5	sk_5	K_5
34	sk_{34}	K_{34}

- What if none of the presented keysets are supported by the card?

When Access is Denied...

ICC

index	RSA	AES
2	pk_2	K_2^{DD}
7	pk_7	K_7^{DD}
$*$	pk^*	K^*

KeySetID = (34, 5) ←

RSA$_{pk^*}$(\$) →

IFD

index	RSA	AES
5	sk_5	K_5
34	sk_{34}	K_{34}

- What if none of the presented keysets are supported by the card?
- The Card will encrypt a randomly generated string using its ShillKey.

When Access is Denied...

ICC

index	RSA	AES
2	pk_2	K_2^{DD}
7	pk_7	K_7^{DD}
$*$	pk^*	K^*

$\xleftarrow{\text{KeySetID} = (34, 5)}$

$\xrightarrow{RSA_{pk^*}(\$)}$

IFD

index	RSA	AES
5	sk_5	K_5
34	sk_{34}	K_{34}

- What if none of the presented keysets are supported by the card?
- The Card will encrypt a randomly generated string using its ShillKey.
- At the IFD side, if no plaintext ending in RND1||RND1 is found, authentication fails (abort).

The PLAID Design and Anonymity

- Recall that in PLAID the RSA encryption keys are kept private.

- The terminal's (inefficient) strategy to sequentially attempt decryption under all of its keys appears to be intended to hide the card's set of keysets, since it could easily be avoided by including the Keyset ID in the clear.

The PLAID Design and Anonymity

- Recall that in PLAID the RSA encryption keys are kept private.
- The terminal's (inefficient) strategy to sequentially attempt decryption under all of its keys appears to be intended to hide the card's set of keysets, since it could easily be avoided by including the Keyset ID in the clear.
- Similarly the Shill key helps to prevent leaking the supported keysets to a probing device.
- The above design factors indicate that PLAID aims to hide a card's set of keysets, i.e. its capabilities.

A Keyset Fingerprinting Attack

A Keyset Fingerprinting Attack

ICC

index	RSA	AES
2	pk_2	K_2^{DD}
7	pk_7	K_7^{DD}

IFD

index	RSA	AES
7	sk_7	K_7
34	sk_{34}	K_{34}

ICC $\longleftarrow$ IFD: KeySetID = (34, 7)

ICC $\longrightarrow$ IFD: $\mathrm{RSA}_{pk_i}(i, \mathrm{DivData}, RND1, RND1)$

ICC $\longleftarrow$ IFD: $\mathrm{AES}_{K_i^{DD}}(\mathrm{AuthReq}, RND2, \mathrm{payload}, k_{\mathrm{session}})$

ICC $\longrightarrow$ IFD: $\mathrm{AES}_{k_{\mathrm{session}}}(\mathrm{AuthResp}, \mathrm{payload}, \mathit{DivData})$

A Keyset Fingerprinting Attack

ICC

index	RSA	AES
2	pk_2	K_2^{DD}
7	pk_7	K_7^{DD}

KeySetID = (34, 7) ←

$\text{RSA}_{pk_7}(7, \text{DivData}, RND1, RND1)$ →

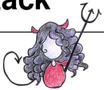
IFD

index	RSA	AES
7	sk_7	K_7
34	sk_{34}	K_{34}

A Keyset Fingerprinting Attack

ICC

2	pk_2	K_2^{DD}
7	pk_7	K_7^{DD}



KeySetID = (34, 7)

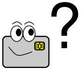
messages are not authenticated!

$\mathrm{RSA}_{pk_7}(7, \mathrm{DivData}, RND1, RND1)$

IFD

index	RSA	AES
7	sk_7	K_7
34	sk_{34}	K_{34}

A Keyset Fingerprinting Attack



KeySetID = (34) ←

ICC

index	RSA	AES
2	pk_2	K_2^{DD}
7	pk_7	K_7^{DD}

IFD

index	RSA	AES
7	sk_7	K_7
34	sk_{34}	K_{34}

Attack Phase 1

- Pick one Keyset ID in the first message and remove all others.

A Keyset Fingerprinting Attack

KeySetID = (34) ←

$\text{RSA}_{pk*}(\$)$ →

ICC

index	RSA	AES
2	pk_2	K_2^{DD}
7	pk_7	K_7^{DD}
$*$	pk^*	K^*

IFD

index	RSA	AES
7	sk_7	K_7
34	sk_{34}	K_{34}

Attack Phase 1

- Pick one Keyset ID in the first message and remove all others.
- Card uses either the listed key or the ShillKey

A Keyset Fingerprinting Attack

ICC

index	RSA	AES
2	pk_2	K_2^{DD}
7	pk_7	K_7^{DD}
$*$	pk^*	K^*

KeySetID = (34) ←

→ $\text{RSA}_{pk^*}(\$)$

IFD

index	RSA	AES
7	sk_7	K_7
34	sk_{34}	K_{34}

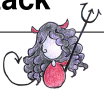
Attack Phase 1

- Pick one Keyset ID in the first message and remove all others.
- Card uses either the listed key or the ShillKey ⇒ check whether the terminal responds with a third message.

A Keyset Fingerprinting Attack

ICC

index	RSA	AES
2	pk_2	K_2^{DD}
7	pk_7	K_7^{DD}
$*$	pk^*	K^*



KeySetID = (34) ←

$\text{RSA}_{pk^*}(\$)$ →

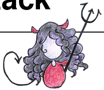
IFD

index	RSA	AES
7	sk_7	K_7
34	sk_{34}	K_{34}

Attack Phase 1

- Pick one Keyset ID in the first message and remove all others.
- Card uses either the listed key or the ShillKey ⇒ check whether the terminal responds with a third message.
- Repeat for all other keysets in the original set

A Keyset Fingerprinting Attack



KeySetID = (34) ←

$RSA_{pk^*}(\$)$ →

ICC

index	RSA	AES
2	pk_2	K_2^{DD}
7	pk_7	K_7^{DD}
*	pk^*	K^*

IFD

index	RSA	AES
7	sk_7	K_7
34	sk_{34}	K_{34}

Attack Phase 1

- Pick one Keyset ID in the first message and remove all others.
- Card uses either the listed key or the ShillKey ⇒ check whether the terminal responds with a third message.
- Repeat for all other keysets in the original set ⇒ determine all supported keysets in the original set.

A Keyset Fingerprinting Attack

KeySetID = (2, 34, 7)

ICC

index	RSA	AES
2	pk_2	K_2^{DD}
7	pk_7	K_7^{DD}
$*$	pk^*	K^*

IFD

index	RSA	AES
7	sk_7	K_7
34	sk_{34}	K_{34}

Attack Phase 2

- Prepend the original set in the first message with a newKeyset ID.

A Keyset Fingerprinting Attack

KeySetID = (2, 34, 7)

ICC

index	RSA	AES
2	pk_2	K_2^{DD}
7	pk_7	K_7^{DD}
$*$	pk^*	K^*

$\mathrm{RSA}_{pk_2}(2, \mathrm{DivData}, RND1, RND1)$

IFD

index	RSA	AES
7	sk_7	K_7
34	sk_{34}	K_{34}

Attack Phase 2

- Prepend the original set in the first message with a newKeyset ID.
- If the new keyset is supported then the terminal will not be able to decrypt it

A Keyset Fingerprinting Attack

ICC

index	RSA	AES
2	pk_2	K_2^{DD}
7	pk_7	K_7^{DD}
*	pk^*	K^*

KeySetID = (2, 34, 7)

$\longleftarrow$

$\mathrm{RSA}_{pk_2}(2, \mathrm{DivData}, RND1, RND1)$

$\longrightarrow$

IFD

index	RSA	AES
7	sk_7	K_7
34	sk_{34}	K_{34}

Attack Phase 2

- Prepend the original set in the first message with a newKeyset ID.
- If the new keyset is supported then the terminal will not be able to decrypt it $\Rightarrow$ No third message.

A Keyset Fingerprinting Attack

TECHNISCHE
UNIVERSITÄT
DARMSTADT

KeySetID = (2, 34, 7)

$\longleftarrow$

ICC

index	RSA	AES
2	pk_2	K_2^{DD}
7	pk_7	K_7^{DD}
$*$	pk^*	K^*

$\text{RSA}_{pk_2}(2, \text{DivData}, RND1, RND1)$

$\longrightarrow$

IFD

index	RSA	AES
7	sk_7	K_7
34	sk_{34}	K_{34}

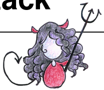
Attack Phase 2

- Prepend the original set in the first message with a newKeyset ID.
- If the new keyset is supported then the terminal will not be able to decrypt it $\Rightarrow$ No third message.
- Repeat for all keysets NOT in the original set

A Keyset Fingerprinting Attack

ICC

index	RSA	AES
2	pk_2	K_2^{DD}
7	pk_7	K_7^{DD}
$*$	pk^*	K^*



KeySetID = (2, 34, 7) ←

$\mathrm{RSA}_{pk_2}(2, \mathrm{DivData}, RND1, RND1)$ →

IFD

index	RSA	AES
7	sk_7	K_7
34	sk_{34}	K_{34}

Attack Phase 2

- Prepend the original set in the first message with a newKeyset ID.
- If the new keyset is supported then the terminal will not be able to decrypt it $\Rightarrow$ No third message.
- Repeat for all keysets NOT in the original set $\Rightarrow$ determine all supported keysets.

Tracing Cards

- In RSA even if the encryption key is kept secret, ciphertexts still **leak a small amount of information** about the encryption key.

- Ciphertexts produced under different keys are distributed differently according to the RSA modulus (e is usually fixed) .

Tracing Cards

- In RSA even if the encryption key is kept secret, ciphertexts still **leak a small amount of information** about the encryption key.
- Ciphertexts produced under different keys are distributed differently according to the RSA modulus (*e* is usually fixed) .
- The RSA Shill Key is generated randomly during the card's initialisation and is essentially **unique** to that card.
- Moreover we can easily sample encryptions under the Shill Key by probing a card with an empty set of Keyset IDs.

Estimating the RSA modulus

- It is reasonable to assume ciphertexts are uniformly distributed over $[0, N-1]$, where N is the modulus.

Estimating the RSA modulus

- It is reasonable to assume ciphertexts are uniformly distributed over $[0, N-1]$, where N is the modulus.
- A naive estimate of the modulus would be to take twice the mean value of the ciphertext samples.

Estimating the RSA modulus

- It is reasonable to assume ciphertexts are uniformly distributed over $[0, N-1]$, where N is the modulus.
- This turns out to be a well studied statistical problem known as the **German tank problem**, due to its application in WWII to estimate the number of German tanks.

Estimating the RSA modulus

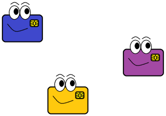
- It is reasonable to assume ciphertexts are uniformly distributed over $[0, N-1]$, where N is the modulus.
- This turns out to be a well studied statistical problem known as the **German tank problem**, due to its application in WWII to estimate the number of German tanks.

$$\tilde{M} = m + \frac{m}{k} - 1$$

- $\tilde{M}$ = Estimated maximum.
- m = Sampled maximum value.
- k = No of samples.

ShillKey Fingerprinting – Scenario 1

ShillKey Fingerprinting – Scenario 1



PLAID system

Attacker

ShillKey Fingerprinting – Scenario 1

ICC$_1$

KeySetID = (“ ”) ⟵

$\text{RSA}_{pk_1^*}(\$)$ ⟶

Attacker

ShillKey Fingerprinting – Scenario 1

ICC_1

Attacker

KeySetID = (“ ”) ←

$\mathrm{RSA}_{pk_1^*}(\$)$ →

- Phase 1 – Identification Phase:
  - for every card i receive k_1 encryptions $\mathrm{RSA}_{pk_i^*}(\$)$

ShillKey Fingerprinting – Scenario 1

ICC_1

$pk_1^* = (N_1, e_1)$

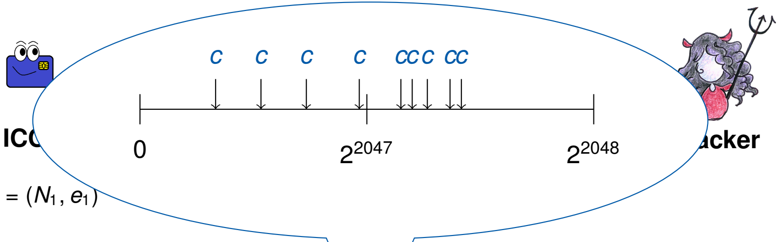
KeySetID = (“ ”) $\longleftarrow$

$1 \leq RSA_{pk_1^*}(\$) < N_1$ $\longrightarrow$

Attacker

- Phase 1 – Identification Phase:
  - for every card i receive k_1 encryptions $RSA_{pk_i^*}(\$)$
  - estimate N_i according to samples

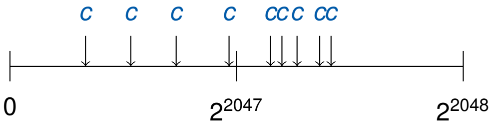
ShillKey Fingerprinting – Scenario 1



- Phase 1 – Identification Phase:
  - for every card i receive k_1 encryptions $\text{RSA}_{pk_i^*}(\$)$
  - estimate N_i according to samples

ShillKey Fingerprinting – Scenario 1

ICC



acker

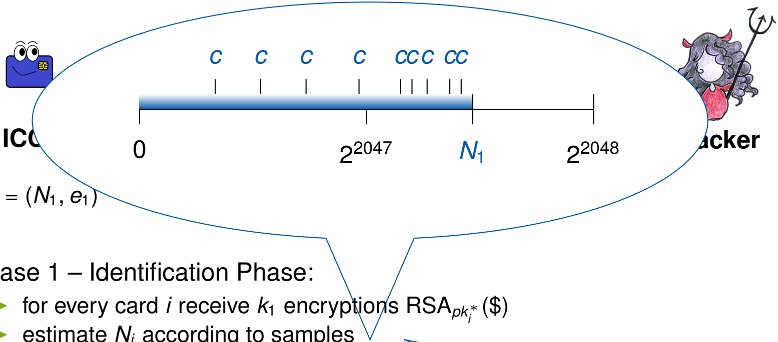
$pk_1^* = (N_1, e_1)$

- Phase 1 – Identification Phase:
  - for every card i receive k_1 encryptions $\mathrm{RSA}_{pk_i^*}(\$)$
  - estimate N_i according to samples

"German Tank Estimate"

$$N_i = \max C + \frac{\max C}{k_1}$$

ShillKey Fingerprinting – Scenario 1



$pk_1^* = (N_1, e_1)$

- Phase 1 – Identification Phase:
  - for every card i receive k_1 encryptions $\mathrm{RSA}_{pk_i^*}(\$)$
  - estimate N_i according to samples

"German Tank Estimate"

$$N_i = \max C + \frac{\max C}{k_1}$$

ShillKey Fingerprinting – Scenario 1

ICC$_1$

$pk_1^* = (N_1, e_1)$

KeySetID = (“ ”)

$1 \leq \mathrm{RSA}_{pk_1^*}(\$) < N_1$

Attacker

N_1

- Phase 1 – Identification Phase:
  - for every card i receive k_1 encryptions $\mathrm{RSA}_{pk_i^*}(\$)$
  - estimate N_i according to samples

ShillKey Fingerprinting – Scenario 1

ICC$_2$

$pk_2^* = (N_2, e_2)$

KeySetID = (“ ”) ←

$\mathrm{RSA}_{pk_2^*}(\$)$ →

Attacker

N_1 N_2

- Phase 1 – Identification Phase:
  - for every card i receive k_1 encryptions $\mathrm{RSA}_{pk_i^*}(\$)$
  - estimate N_i according to samples

ShillKey Fingerprinting – Scenario 1

ICC$_3$

$pk_3^* = (N_3, e_3)$

KeySetID = (“ ”) ←

RSA$_{pk_3^*}$($) →

Attacker

N_1 N_2 N_3

- Phase 1 – Identification Phase:
  - for every card i receive k_1 encryptions RSA$_{pk_i^*}$($)
  - estimate N_i according to samples

ShillKey Fingerprinting – Scenario 1

Attacker

N_1 N_2 N_3

- Phase 1 – Identification Phase:
  - for every card i receive k_1 encryptions $\mathrm{RSA}_{pk_i^*}(\$)$
  - estimate N_i according to samples
- Phase 2 – Challenge Phase:

ShillKey Fingerprinting – Scenario 1

$ICC_?$

$pk^* = (N^*, e^*)$

KeySetID = (“ ”) ←

$\mathrm{RSA}_{pk^*}(\$)$ →

Attacker

N_1 N_2 N_3

- Phase 1 – Identification Phase:
  - for every card i receive k_1 encryptions $\mathrm{RSA}_{pk_i^*}(\$)$
  - estimate N_i according to samples
- Phase 2 – Challenge Phase:
  - receive k_2 encryptions $\mathrm{RSA}_{pk^*}(\$)$

TECHNISCHE UNIVERSITÄT DARMSTADT

ShillKey Fingerprinting – Scenario 1

$ICC_?$

$pk^* = (N^*, e^*)$

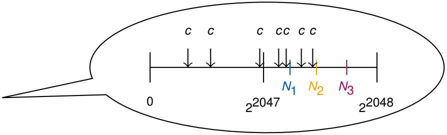
KeySetID = (“ ”) ←

$\mathrm{RSA}_{pk^*}(\$)$ →

Attacker

N_1 N_2 N_3

- Phase 1 – Identification Phase:
  - for every card i receive k_1 encryptions $\mathrm{RSA}_{pk_i^*}(\$)$
  - estimate N_i according to samples
- Phase 2 – Challenge Phase:
  - receive k_2 encryptions $\mathrm{RSA}_{pk^*}(\$)$
  - estimate N^* as in Phase 1



TECHNISCHE UNIVERSITÄT DARMSTADT

ShillKey Fingerprinting – Scenario 1

ICC$_2$

$pk_2^* = (N_2, e_2)$

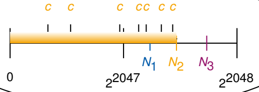
KeySetID = (“ ”) ←

$RSA_{pk_2^*}(\$)$ →

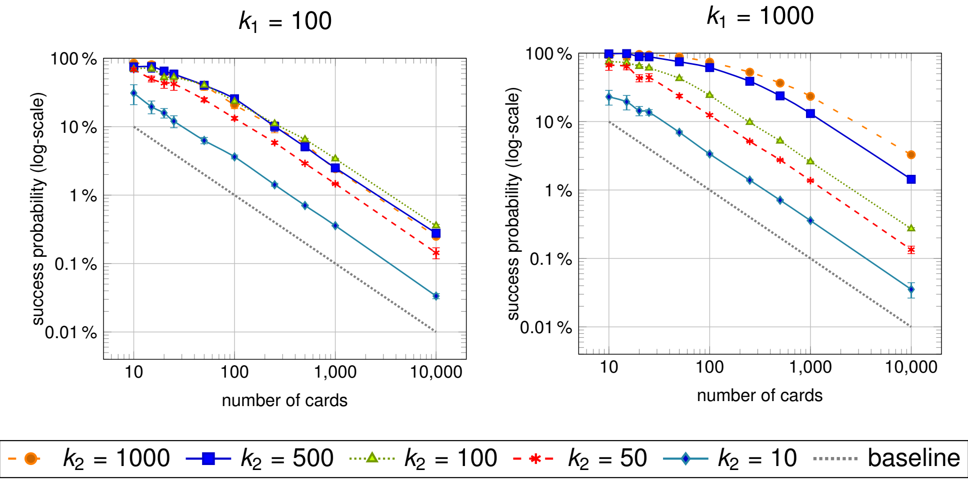
Attacker

N_1 N_2 N_3

- Phase 1 – Identification Phase:
  - for every card i receive k_1 encryptions $RSA_{pk_i^*}(\$)$
  - estimate N_i according to samples
- Phase 2 – Challenge Phase:
  - receive k_2 encryptions $RSA_{pk^*}(\$)$
  - estimate N^* as in Phase 1
  - guess card j with $\min_j |N^* - N_j|$



ShillKey Fingerprinting – Scenario 1 – Results



ShillKey Fingerprinting – Scenario 2

- In the previous scenario we had the ability to interact k_1 times with each card, which may not always be realistic.
- We now consider a setting where we are given a **mixed set** of ciphertexts, without knowing which ciphertexts come from the same key.

ShillKey Fingerprinting – Scenario 2

- In the previous scenario we had the ability to interact k_1 times with each card, which may not always be realistic.
- We now consider a setting where we are given a **mixed set** of ciphertexts, without knowing which ciphertexts come from the same key.
- This scenario can arise for instance if the attacker manages to install a **fake terminal** or to '**skim**' a terminal.

ShillKey Fingerprinting – Scenario 1

Let t = Number of cards in the system.

- Phase 1 – Identification Phase:
  - for every card i receive k_1 encryptions $\text{RSA}_{pk_i^*}(\$)$
  - estimate N_i according to samples.
- Phase 2 – Challenge Phase:
  - receive k_2 encryptions $\text{RSA}_{pk^*}(\$)$
  - estimate N^* from the k_2 samples.
  - guess card j with $\min_j |N^* - N_j|$.

ShillKey Fingerprinting – Scenario 2

Let t = Number of cards in the system.

- Phase 1 – Identification Phase:
  - receive $k_1 \cdot t$ random samples $\text{RSA}_{pk*}(\$)$
  - estimate N_i according to samples.

- Phase 2 – Challenge Phase:
  - receive k_2 encryptions $\text{RSA}_{pk*}(\$)$
  - estimate N^* from the k_2 samples.
  - guess card j with $\min_j |N^* - N_j|$.

ShillKey Fingerprinting – Scenario 2

Let t = Number of cards in the system.

- Phase 1 – Identification Phase:
  - receive $k_1 \cdot t$ random samples $\mathrm{RSA}_{pk^*}(\$)$
  - estimate N_i according to samples.
- Phase 2 – Challenge Phase:
  - receive k_2 encryptions $\mathrm{RSA}_{pk^*}(\$)$
  - estimate N^* from the k_2 samples.
  - guess card j with $\min_j |N^* - N_j|$.

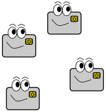
ShillKey Fingerprinting – Scenario 2

Let t = Number of cards in the system.

- Phase 1 – Identification Phase:
  - receive $k_1 \cdot t$ random samples $\mathrm{RSA}_{pk*}(\$)$
  - estimate N_i according to samples.
- Phase 2 – Challenge Phase:
  - receive k_2 encryptions $\mathrm{RSA}_{pk*}(\$)$
  - estimate N^* from the k_2 samples.
  - guess card j with $\min_j |N^* - N_j|$.

We use a heuristic clustering technique from machine learning to sort the ciphertext samples, and then get an estimate from each cluster.

ShillKey Fingerprinting – Scenario 2



ICC$_?$

$pk_?^* = (N_?, e_?)$

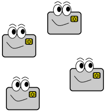
$$\xleftarrow{\text{KeySetID = (“ ”)}}$$

$$\xrightarrow{\mathrm{RSA}_{pk_?^*}(\$)}$$

?

Attacker

ShillKey Fingerprinting – Scenario 2



ICC$_?$

KeySetID = (“ ”) $\longleftarrow$

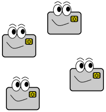
?

Attacker

$pk_?^* = (N_?, e_?)$ $\xrightarrow{\text{RSA}_{pk_?^*}(\$)}$

- standard clustering technique based on *k*-means algorithm

ShillKey Fingerprinting – Scenario 2



ICC$_?$

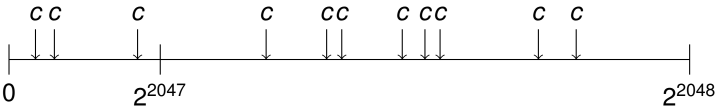
$pk_?^* = (N_?, e_?)$

KeySetID = (“ ”) $\longleftarrow$

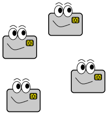
$\mathrm{RSA}_{pk_?^*}(\$)$ $\longrightarrow$

Attacker

- standard clustering technique based on k-means algorithm



ShillKey Fingerprinting – Scenario 2



ICC$_?$

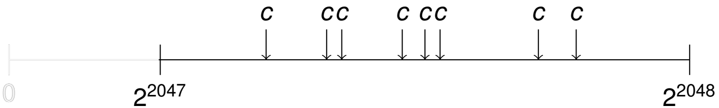
$pk_?^* = (N_?, e_?)$

KeySetID = (“ ”) ←

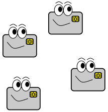
$\mathrm{RSA}_{pk_?^*}(\$)$ →

Attacker

- standard clustering technique based on k-means algorithm



ShillKey Fingerprinting – Scenario 2



$ICC_?$

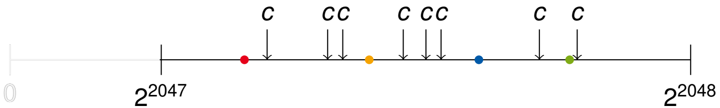
$pk_?^* = (N_?, e_?)$

KeySetID = (“ ”) ←

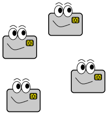
$RSA_{pk_?^*}(\$)$ →

Attacker

- standard clustering technique based on k-means algorithm



ShillKey Fingerprinting – Scenario 2



ICC$_?$

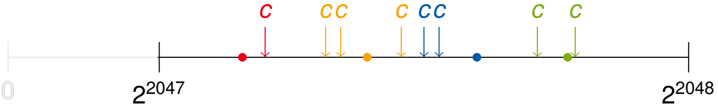
KeySetID = (“ ”) $\longleftarrow$

Attacker

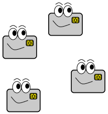
$\mathrm{RSA}_{pk_?^*}(\$)$ $\longrightarrow$

$pk_?^* = (N_?, e_?)$

- standard clustering technique based on k-means algorithm



ShillKey Fingerprinting – Scenario 2



ICC?

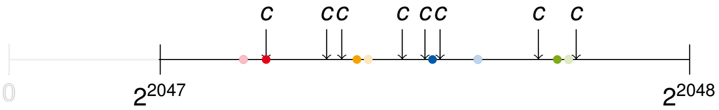
KeySetID = (" ")

Attacker

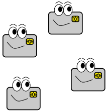
$\mathrm{RSA}_{pk^*_?}(\$)$

$pk^*_? = (N_?, e_?)$

- standard clustering technique based on k-means algorithm



ShillKey Fingerprinting – Scenario 2



$\mathrm{ICC}_?$

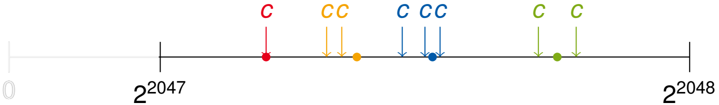
$pk^*_? = (N_?, e_?)$

$\xleftarrow{\text{KeySetID = (" ")}}$

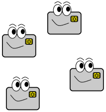
$\xrightarrow{\mathrm{RSA}_{pk^*_?}(\$)}$

Attacker

- standard clustering technique based on k-means algorithm



ShillKey Fingerprinting – Scenario 2



ICC$_?$

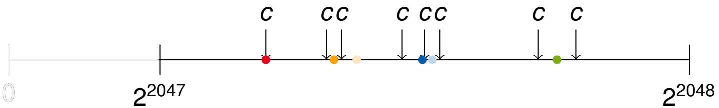
$pk_?^* = (N_?, e_?)$

KeySetID = (“ ”) ⟵

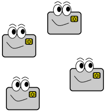
RSA$_{pk_?^*}$ ($) ⟶

Attacker

- standard clustering technique based on k-means algorithm



ShillKey Fingerprinting – Scenario 2



ICC$_?$

$pk^*_? = (N_?, e_?)$

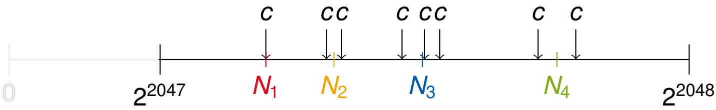
KeySetID = (“ ”) ←

RSA$_{pk^*_?}$ (\$) →

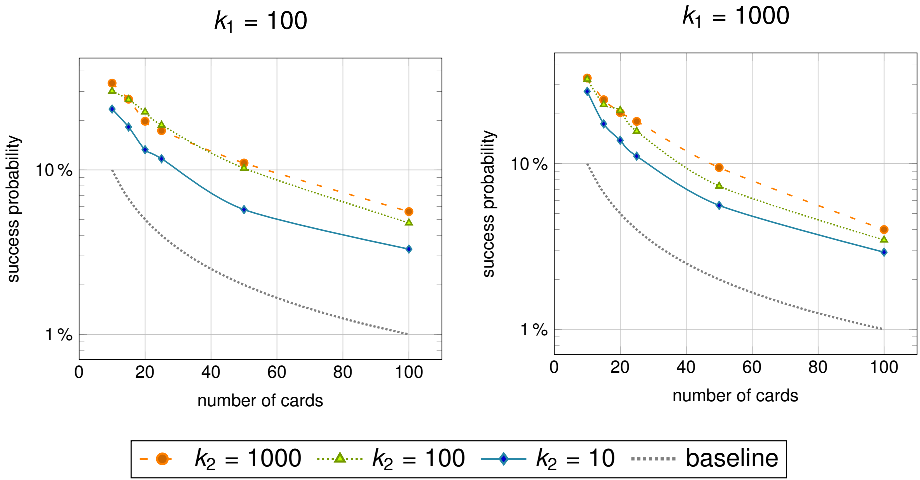
Attacker

N_1 N_2 N_3 N_4

- standard clustering technique based on k-means algorithm



ShillKey Fingerprinting – Scenario 2 – Results



ShillKey Fingerprinting – Scenario 3

- We now further restrict the identification phase to only obtain k_1 ciphertexts from only one target card.
- In the challenge phase we will be given k_2 ciphertexts coming either from the target card or a randomly generated card. The **goal** is to distinguish the two.
- Note that while the challenge phase looks simpler, it is also the case that now we have no information about the other cards to aid the challenge phase.

ShillKey Fingerprinting – Scenario 3

- Phase 1 – Identification Phase:
  - receive k_1 encryptions $\mathrm{RSA}_{pk_t^*}(\$)$ from a target card.
  - estimate N_t using the GTE.

ShillKey Fingerprinting – Scenario 3

- Phase 1 – Identification Phase:
  - receive k_1 encryptions $\mathrm{RSA}_{pk_t^*}(\$)$ from a target card.
  - estimate N_t using the GTE.
  - estimate the **variance** of N_t.

$$\sigma^2 = \frac{1}{k} \cdot \frac{(N-k)(N+1)}{k+2}$$

ShillKey Fingerprinting – Scenario 3

- Phase 1 – Identification Phase:
  - receive k_1 encryptions $\mathrm{RSA}_{pk_t^*}(\$)$ from a target card.
  - estimate N_t using the GTE.
  - estimate the **variance** of N_t.
- Phase 2 – Challenge Phase:
  - receive k_2 encryptions $\mathrm{RSA}_{pk^*}(\$)$ from on one card.
  - estimate N^* using GTE.

$$\sigma^2 = \frac{1}{k} \cdot \frac{(N-k)(N+1)}{k+2} \qquad k = \min(k_1, k_2)$$

ShillKey Fingerprinting – Scenario 3

- Phase 1 – Identification Phase:
  - receive k_1 encryptions $\mathrm{RSA}_{pk_t^*}(\$)$ from a target card.
  - estimate N_t using the GTE.
  - estimate the **variance** of N_t.

- Phase 2 – Challenge Phase:
  - receive k_2 encryptions $\mathrm{RSA}_{pk^*}(\$)$ from on one card.
  - estimate N^* using GTE.
  - guess card is the target card iff $|N^* - N_t| < 3\sigma$

$$\sigma^2 = \frac{1}{k} \cdot \frac{(N-k)(N+1)}{k+2}$$

$$k = \min(k_1, k_2)$$

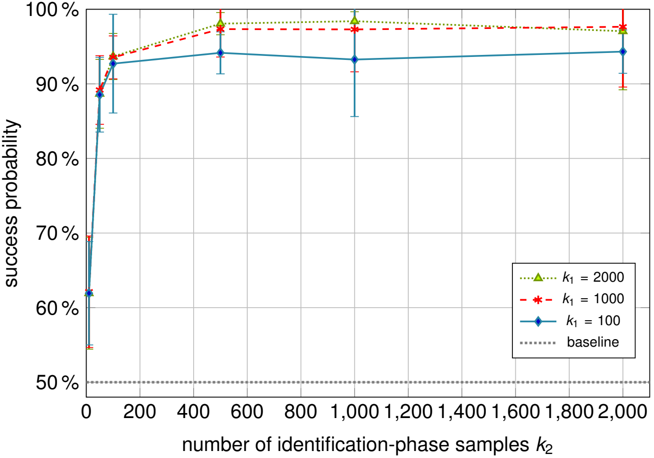
FRR = 2%

$k = 100 \rightarrow \mathrm{FAR} = 5\%,$

$k = 1000 \rightarrow \mathrm{FAR} = 0.5\%$

ShillKey Fingerprinting – Scenario 1 – Results

TECHNISCHE UNIVERSITÄT DARMSTADT



PLAID


PLAID

Other Issues with PLAID

- Remember that at the end of a PLAID protocol run the card and the terminal share a **session key**.

- **No Forward security**: a compromise of the long-term keys of either party, immediately results in a compromise of past session keys.

Other Issues with PLAID

TECHNISCHE UNIVERSITÄT DARMSTADT

- Remember that at the end of a PLAID protocol run the card and the terminal share a **session key**.

- **No Forward security**: a compromise of the long-term keys of either party, immediately results in a compromise of past session keys.

- For RSA, PLAID uses PKCS#1 v1.5 instead of OAEP, which is widely known to be vulnerabe to Bleichenbacher's attack.

TECHNISCHE UNIVERSITÄT DARMSTADT

Other Issues with PLAID

- Remember that at the end of a PLAID protocol run the card and the terminal share a **session key**.

- **No Forward security**: a compromise of the long-term keys of either party, immediately results in a compromise of past session keys.

- For RSA, PLAID uses PKCS#1 v1.5 instead of OAEP, which is widely known to be vulnerabe to Bleichenbacher's attack.

- While we didn't see a direct way of exploiting it, the designers claim that Bleichenbacher's attack does not apply to PLAID simply because the RSA moduli are not public!.

Other Issues with PLAID

- For symmetric encryption PLAID uses AES in CBC mode with a **fixed IV of zeros**.
- Thus encryption is deterministic and therefore not IND-CPA secure.

Other Issues with PLAID

- For symmetric encryption PLAID uses AES in CBC mode with a **fixed IV of zeros**.
- Thus encryption is deterministic and therefore not IND-CPA secure.
- The CBC padding is based on ISO/IEC 9797-1, but is incorrectly specified so that it is not uniquely decodable.

Other Issues with PLAID

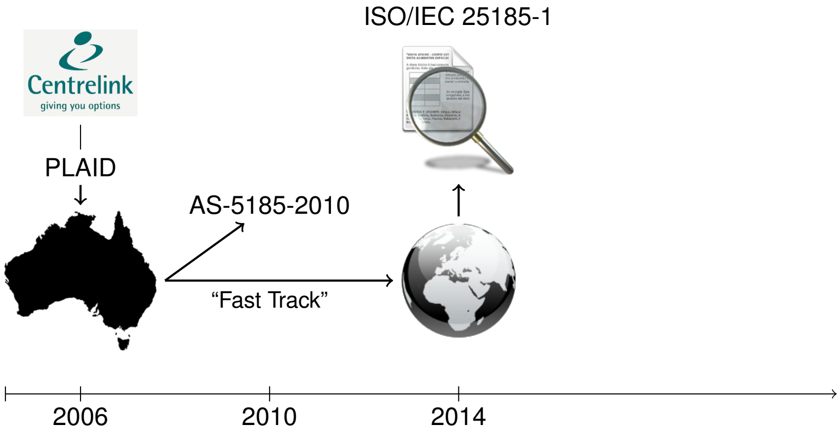
- For symmetric encryption PLAID uses AES in CBC mode with a **fixed IV of zeros**.
- Thus encryption is deterministic and therefore not IND-CPA secure.
- The CBC padding is based on ISO/IEC 9797-1, but is incorrectly specified so that it is not uniquely decodable.
- No authentication (MAC) is applied to CBC encryption.

Other Issues with PLAID

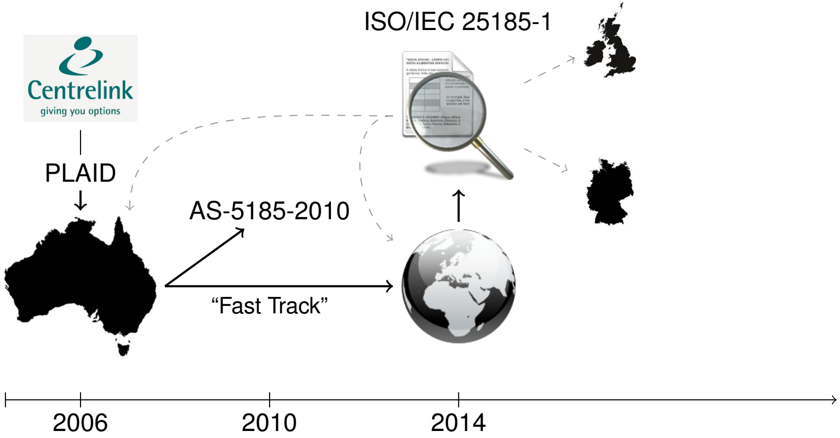
- For symmetric encryption PLAID uses AES in CBC mode with a **fixed IV of zeros**.
- Thus encryption is deterministic and therefore not IND-CPA secure.
- The CBC padding is based on ISO/IEC 9797-1, but is incorrectly specified so that it is not uniquely decodable.
- No authentication (MAC) is applied to CBC encryption.
- The list goes on....

Timeline

Timeline

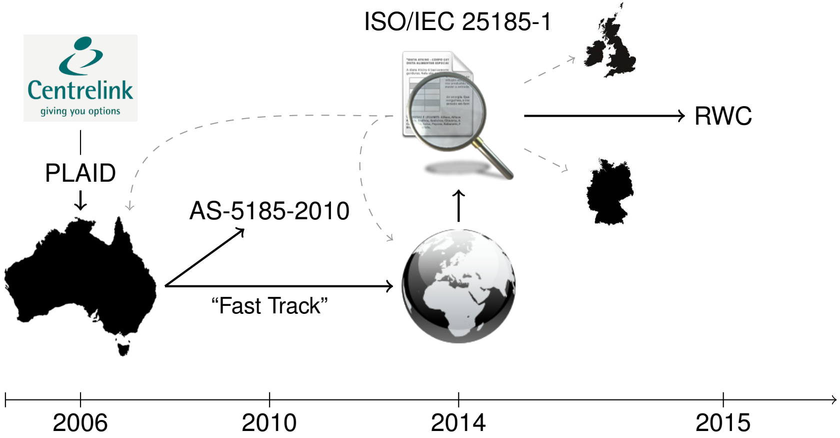

ISO/IEC 25185-1
Centrelink
giving you options
PLAID
AS-5185-2010
“Fast Track”
2006
2010
2014

Timeline

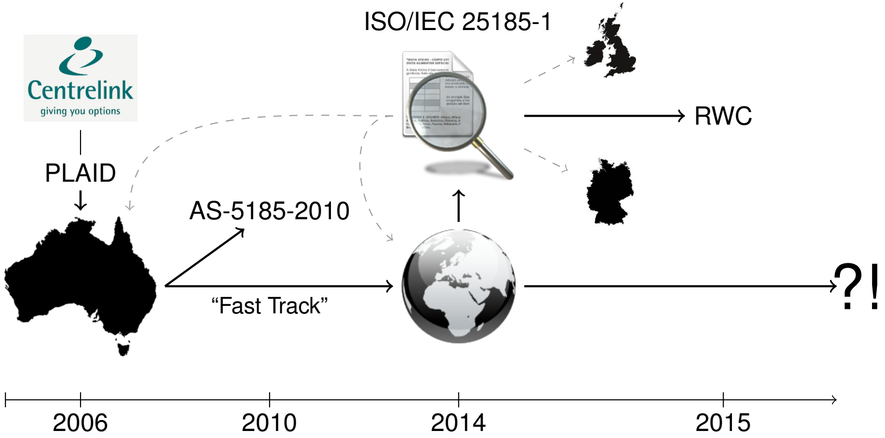


ISO/IEC 25185-1
Centrelink
giving you options
PLAID
AS-5185-2010
"Fast Track"
2006
2010
2014

Timeline

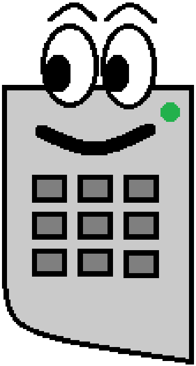

Centrelink
giving you options
PLAID
AS-5185-2010
“Fast Track”
ISO/IEC 25185-1
RWC
2006
2010
2014
2015

Timeline


Centrelink
giving you options
PLAID
AS-5185-2010
“Fast Track”
ISO/IEC 25185-1
RWC
?!
2006
2010
2014
2015

Thank you for your attention!

TECHNISCHE UNIVERSITÄT DARMSTADT



ROYAL HOLLOWAY UNIVERSITY OF LONDON

Information Security Group

EPSRC Engineering and Physical Sciences Research Council

TECHNISCHE UNIVERSITÄT DARMSTADT

Cryptoplexity Cryptography & Complexity Theory Technische Universität Darmstadt www.cryptoplexity.de

CROSSING

CASED

Heisenberg-Programm Deutsche Forschungsgemeinschaft

EC SPRIDE EUROPEAN CENTER FOR SECURITY AND PRIVACY BY DESIGN