



UNIVERSITY OF
SURREY



Fine-Grained Trackability in Protocol Executions

NDSS 2023

Ksenia Budykho,
Ioana Boureanu,
Stephan Wesemeyer
Surrey Centre for Cyber Security,
University of Surrey, UK
sccs@surrey.ac.uk

Daniel Romero,
Matt Lewis
NCC Group, UK
daniel.romero@nccgroup.com,
matt.lewis@nccgroup.com

Yogaratnam Rahulan
5G/6G Innovation Centre,
University of Surrey, UK
y.rahulan@surrey.ac.uk

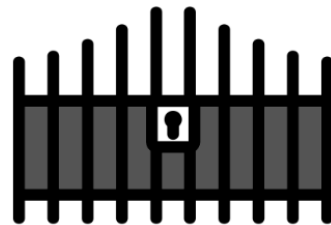
Fortunat Rajaona,
Steve Schneider
Surrey Centre for Cyber Security,
University of Surrey, UK
sccs@surrey.ac.uk

What is trackability?

What is trackability?

Security vs Privacy

- Security: protection against information disclosure, theft of electronic data, protection of hardware
- Privacy: protection of namelessness
- Anonymity, unlinkability, untraceability



What is trackability?

- New: trackability!
 - Link permanent identifier to temporary identifier to track secured application-level traffic
 - Finer-grained than other existing notions
- New: *TrackDev*!
 - Any secure application-level traffic
 - Mechanisable (Tamarin)

The *TrackDev* framework



March 2, 2023

Ksenia Budykho

5



TrackDev Identification Functionality

1. $A \rightarrow B: m_1$



Alice

Hi! I'm Alice!



Bob

TrackDev Identification Functionality

[p. B \rightarrow A: m₂]



Alice



Bob

TrackDev Identification Functionality

q. A \rightarrow B: m_3



“Alice”

Give me the application!



Bob

TrackDev Identification Functionality

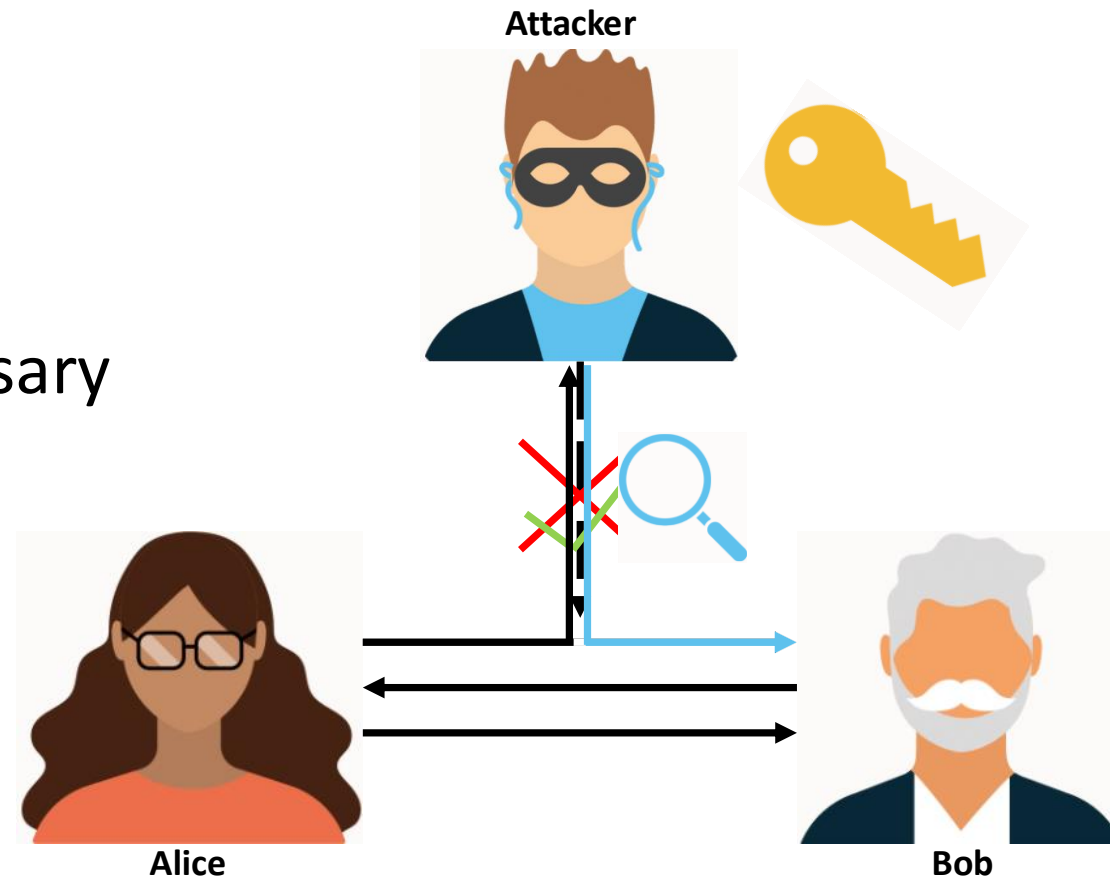
1. $A \rightarrow B: m_1$
- [p. $B \rightarrow A: m_2$]
- q. $A \rightarrow B: m_3$

Req. 1: In steps 1 to (q-1), B identifies A via id_A

Req. 2: In steps q and thereafter, A is accessing an application-level service facilitated by B

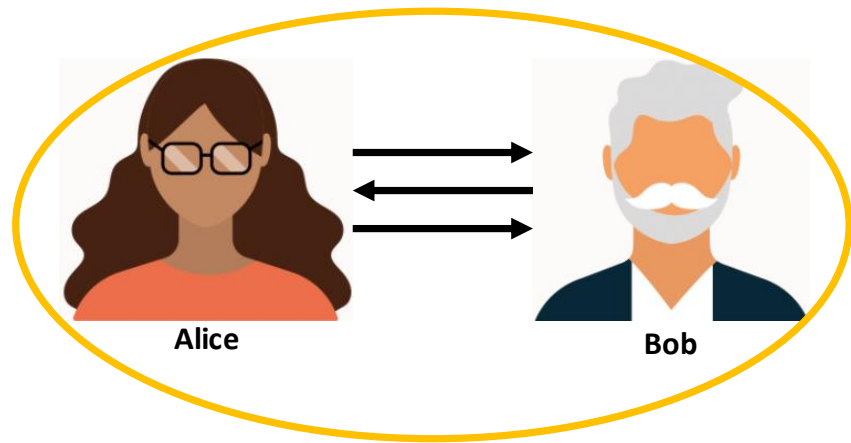
Attackers in *TrackDev*

- Two types
 - Passive (weak)
 - Active (strong)
- Akin to the Dolev Yao adversary
 - Perfect cryptography

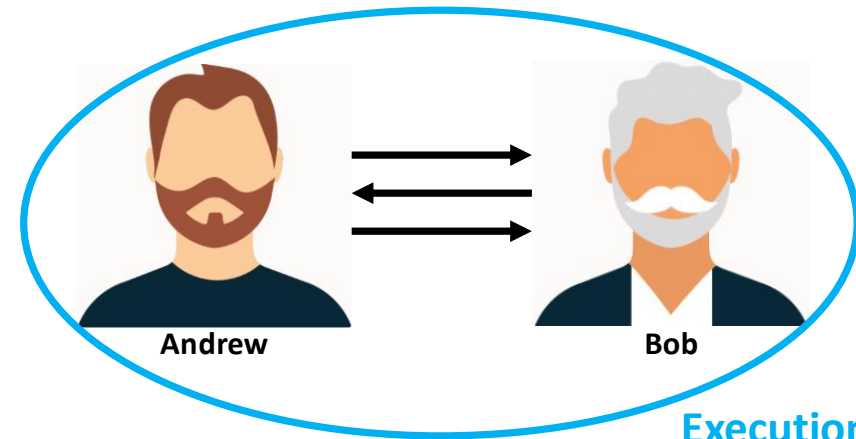


Attack Setup in *TrackDev*

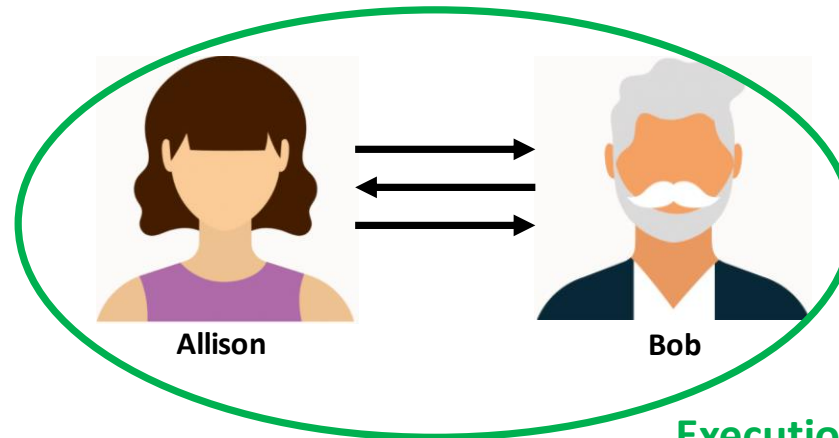
- A set of executions/runs E of protocol



Execution 1



Execution 3

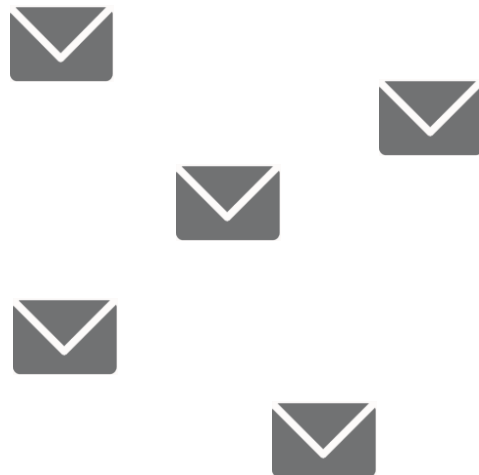


Execution 2

Attack Setup in *TrackDev*

- Attack Setup **S(E)** over that set of executions **E**

Pre-Application Messages



+

Application-Level Messages



Attack Setup in *TrackDev*

- Attack Setup $S(E)$ over that set of executions E

Pre-Application Messages



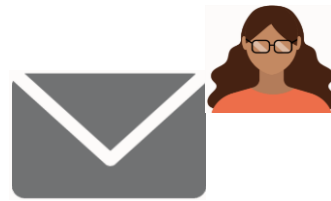
+

Application-Level Messages



A Trackability Attack in *TrackDev*

- Attack Setup $S(E)$
- Trackability Relation $Tr = (id_A, M_A)$
 - Pairs (id_A, M_A) such that:
 - M_A is a set of application messages M_{App} in the attack setup $S(E)$
 - M_A were really sent by A identified as id_A



Attacker



Characteristics of *TrackDev*

Characteristics of *TrackDev*

Explicit vs Implicit Non-Trackability

- **Explicit** – everyone is using their ‘real’ identifiers
- **Implicit** – the identifiers are masked to protect the party’s privacy
 - $\varphi(\text{id}_{\text{Alice}})$



Explicit
Alice

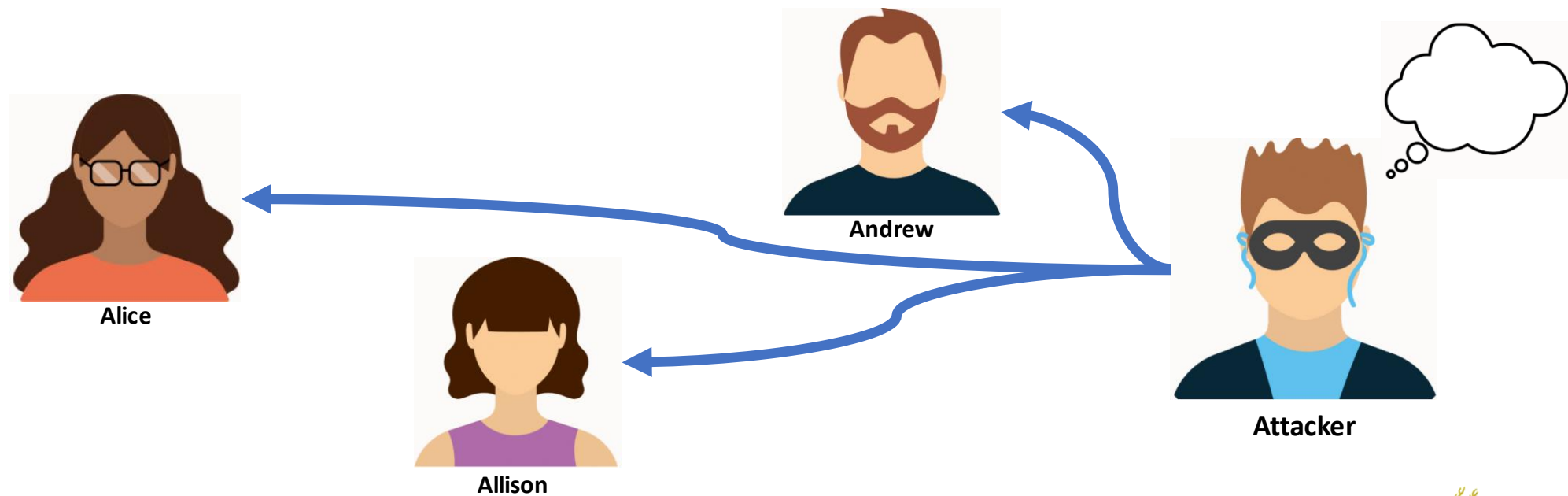


Implicit
“Alice”

Characteristics of *TrackDev*

Static vs Adaptive Non-Trackability

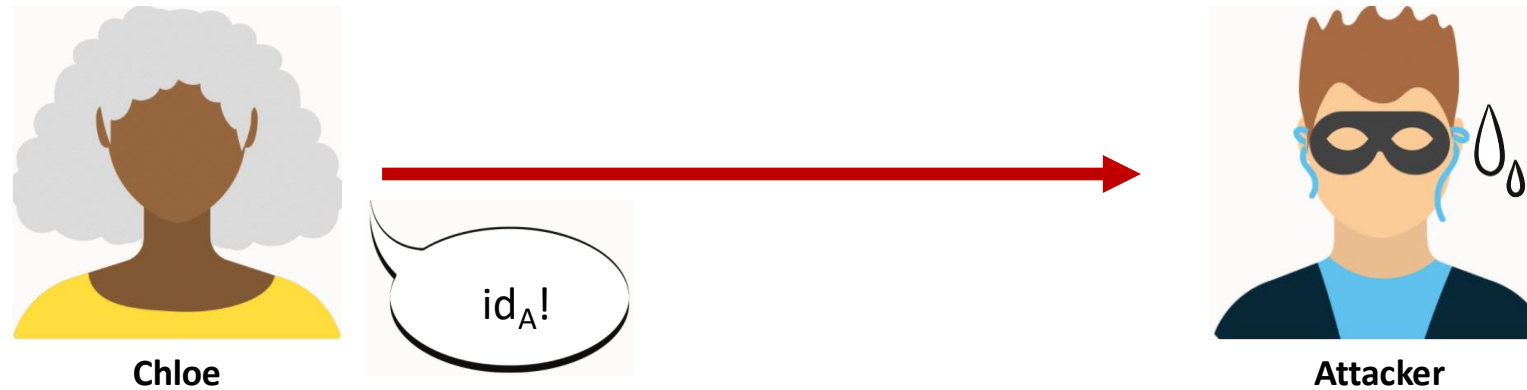
- The identifier(s) to be tracked are not imposed from outside
- Adversary free to choose who they track



Characteristics of *TrackDev*

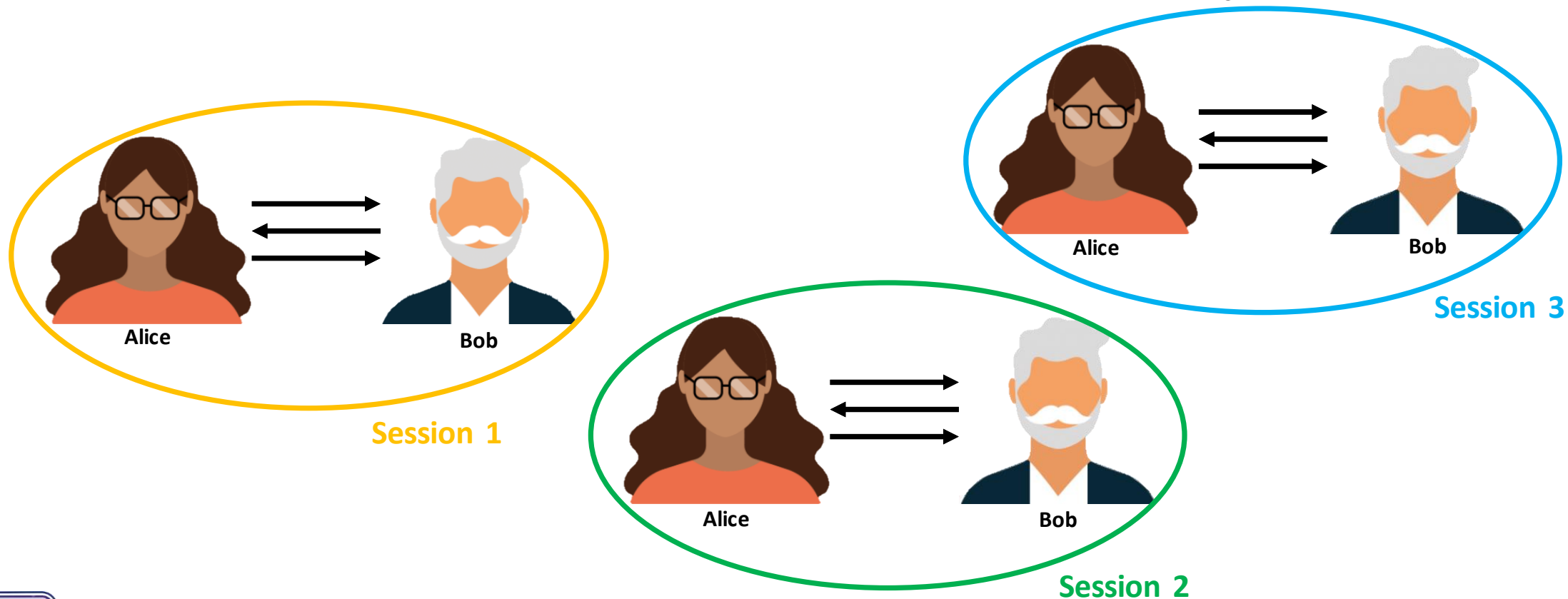
Static vs **Adaptive** Non-Trackability

- The identifier(s) to be tracked are imposed from outside
- Adversary is challenged on a specific party to track



Characteristics of *TrackDev*

Session-Insensitive vs ~~Session-Sensitive~~ Non-Trackability



Characteristics of *TrackDev*

Session-Insensitive vs **Session-Sensitive** Non-Trackability

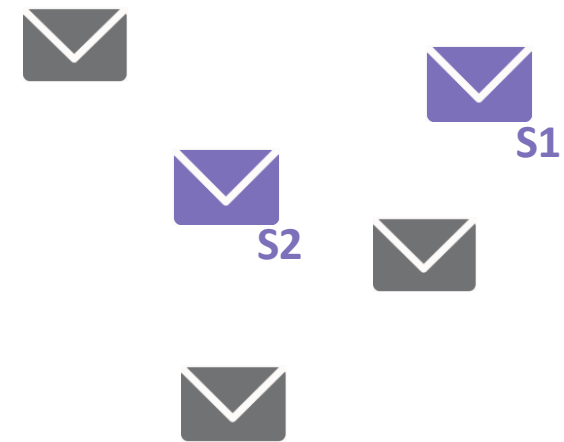
- $S(E, Sess) = (M^{Sess}_{id}, M^{Sess}_{App})$

Pre-Application Messages



+

Application-Level Messages





Characteristics of *TrackDev*

- Active vs Passive attacker
- Explicit vs Implicit
- Static vs Adaptive
- Session-Insensitive vs Session-Sensitive
- Existential vs Universal

Comparison of *TrackDev* characteristics

Comparison of *TrackDev* characteristics

	E-St-NoTrack	E-St-SesNoTrack	V-St-NoTrack	V-St-SesNoTrack	E-Ad-NoTrack	E-Ad-SesNoTrack	V-Ad-NoTrack	V-Ad-SesNoTrack
implicit, passive								
implicit, active								
explicit, passive								
explicit, active								

- Non-Trackability vs Trackability (attack)

Comparison of *TrackDev* characteristics

- *TrackDev*: more granular way of creating privacy specification
- *TrackDev*: generic recipe for lemmas (for explicit trackability)



Implications between *TrackDev* notions



Track^{active} \Leftrightarrow Track^{passive}




Ad-Track \Leftrightarrow St-Track



Other implications

Comparison of *TrackDev* characteristics

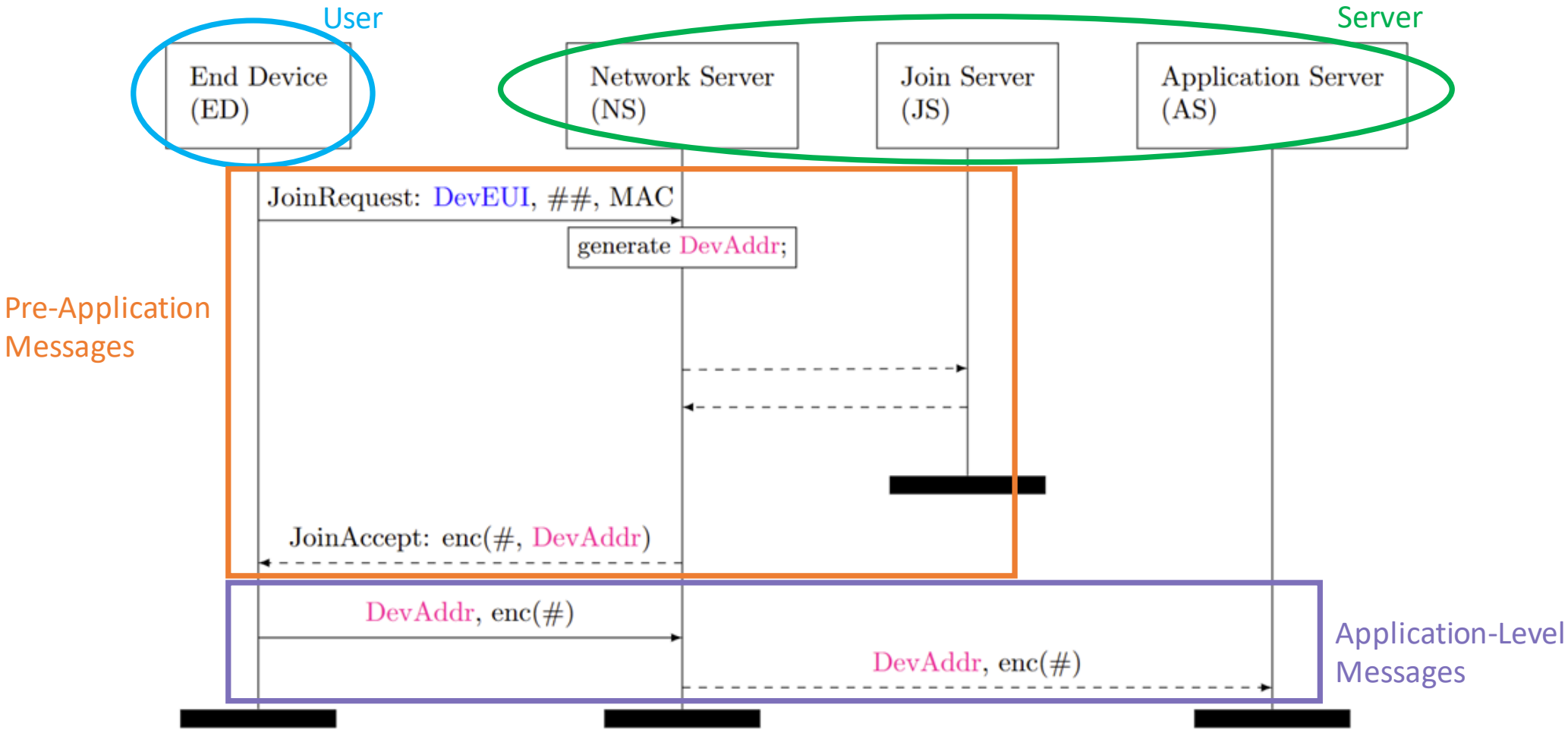
- Attacker
 - Active
 - Passive

Adaptive ^{active}  Static ^{passive}



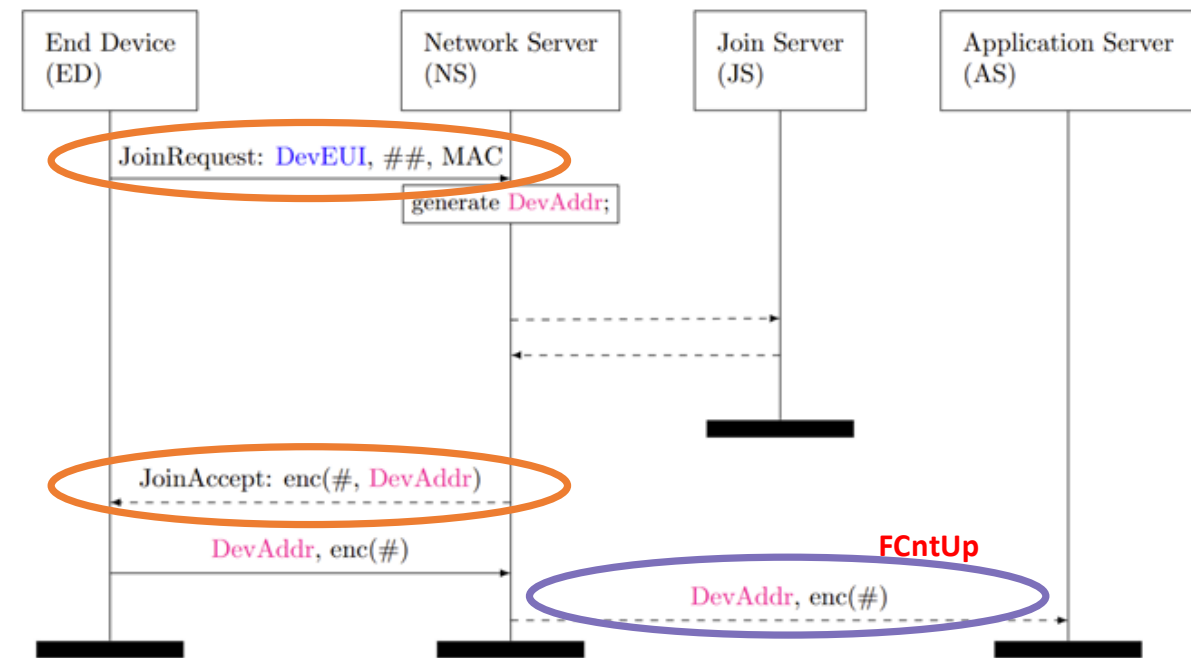
TrackDev examples

TrackDev example: LoRaWAN Join



TrackDev example: LoRaWAN Join

- St-SesTrack^{explicit,passive}
- **n** devices on network
 - Only 1 device joining
- Trackability Relation $\mathbf{Tr} = (\mathbf{id}_{A1}, \mathbf{M}_{A1})$
- $\{ (\text{DevEUI}_1, \{ (\text{DevAddr}_{i1}, \text{data}_{i1}), \dots, (\text{DevAddr}_{i1}, \text{data}_{i1}) \}) \}$



TrackDev in practice

- FLoRa + LoRaWAN Join
- 5G



March 2, 2023

Ksenia Budykho



FloRa + LoRaWAN Join

FLoRa v2.0 Sniffer & Analyzer Send LoRaWAN Packets

* 20210905_113417 All Packets No Attack Selected Only valid packets Application Key Apply Encryption Key Network Key Apply Network Key

ID:	Type:	Reception Time:	Packet info: ⓘ	Decrypted data: ⓘ	Invalid: ⓘ	MIC:	Actions:
<input type="checkbox"/> 1	Downlink Msg (UC)	2021-09-05 13:18:48.44	Freq: 868.5 Chan: 2 Datr: SF7BW125 DevAddr: 260bb3ac ACK: 1 FCnt: 0003				Open Replay
<input type="checkbox"/> 2	Uplink Msg (C)	2021-09-05 13:18:43.40	Freq: 868.5 Chan: 2 Datr: SF7BW125 DevAddr: 260bb3ac ACK: 0 FCnt: 0003				Open Replay
<input type="checkbox"/> 3	Join-Request	2021-09-05 13:18:39.32	Freq: 867.1 Chan: 3 Datr: SF7BW125 JoinEUI: cb954fdad8875540 DevEUI: fd06d2fcee3b578a DevNonce: a96a	N/A (Join-Requests are not encrypted)	Y		Open Replay
<input type="checkbox"/> 4	Downlink Msg (UC)	2021-09-05 13:17:48.45	Freq: 867.5 Chan: 5 Datr: SF7BW125 DevAddr: 260bb3ac ACK: 1 FCnt: 0002				Open Replay
<input type="checkbox"/> 5	Uplink Msg (C)	2021-09-05 13:17:43.40	Freq: 867.5 Chan: 5 Datr: SF7BW125 DevAddr: 260bb3ac ACK: 0 FCnt: 0002				Open Replay
<input type="checkbox"/> 6	Uplink Msg (C)	2021-09-05 13:17:12.82	Freq: 867.1 Chan: 3 Datr: SF7BW125 DevAddr: b057cb9a ACK: 1 FCnt: d12e				Open Replay
<input type="checkbox"/> 7	Downlink Msg (UC)	2021-09-05 13:16:48.48	Freq: 867.7 Chan: 6 Datr: SF7BW125 DevAddr: 260bb3ac ACK: 1 FCnt: 0001				Open Replay
<input type="checkbox"/> 8	Uplink Msg (C)	2021-09-05 13:16:43.41	Freq: 867.7 Chan: 6 Datr: SF7BW125 DevAddr: 260bb3ac ACK: 0 FCnt: 0001				Open Replay
<input type="checkbox"/> 9	Uplink Msg (C)	2021-09-05 13:16:43.41	Freq: 867.3 Chan: 4 Datr: SF7BW125 DevAddr: 260bb3ac ACK: 0 FCnt: 0001				Open Replay
<input type="checkbox"/> 10	Downlink Msg (C)	2021-09-05 13:16:00.76	Freq: 867.7 Chan: 6 Datr: SF12BW125 DevAddr: 260b6778 ACK: 0 FCnt: 0001				Open Replay
<input type="checkbox"/> 11	Uplink Msg (UC)	2021-09-05 13:15:54.28	Freq: 867.3 Chan: 4 Datr: SF12BW125 DevAddr: 260b6778 ACK: 0 FCnt: 0000				Open Replay
<input type="checkbox"/> 12	Uplink Msg (UC)	2021-09-05 13:15:54.28	Freq: 867.7 Chan: 6 Datr: SF12BW125 DevAddr: 260b6778 ACK: 0 FCnt: 0000				Open Replay

FloRa + LoRaWAN Join

- DevEUI₁: 75 C6 00 00 0A CA 25 00
- DevEUI₂: 64 7F DA 00 00 00 3F ED
- St-SesTrack^{explicit,passive}
- Target DevEUI₁
 - DevEUI₁: 75 C6 00 00 0A CA 25 00
 - DevAddr: 26 0B 05 23
 - DevAddr: 26 0B 24 D9

Time	Type	DevAddr	DevEUI	FCnt
'2021-09-05 11:34:23.87'	'Join-Request'		'b6bf6b860a8b1ea7'	
'2021-09-05 11:34:56.31'	'Join-Request'		'75c600000aca2500'	
'2021-09-05 11:35:01.39'	'Join-Accept'			
'2021-09-05 11:35:04.25'	'Uplink Msg (C)'	'260b0523'		'0000'
'2021-09-05 11:35:04.26'	'Uplink Msg (C)'	'260b0523'		'0000'
'2021-09-05 11:35:17.95'	'Join-Request'		'647fda0000003fed'	
'2021-09-05 11:35:40.26'	'Uplink Msg (UC)'	'260be271'		'0000'
'2021-09-05 11:35:40.27'	'Uplink Msg (UC)'	'260be271'		'0000'
'2021-09-05 11:36:04.51'	'Uplink Msg (C)'	'260b0523'		'0001'
'2021-09-05 11:37:04.51'	'Uplink Msg (C)'	'260b0523'		'0002'
'2021-09-05 11:37:53.19'	'Uplink Msg (C)'	'e96adc6f'		'4c47'
'2021-09-05 11:38:04.50'	'Uplink Msg (C)'	'260b0523'		'0003'
'2021-09-05 11:38:04.50'	'Uplink Msg (C)'	'260b0523'		'0003'
'2021-09-05 11:39:00.35'	'Join-Request'		'790ef4364d04e6ab'	
'2021-09-05 11:39:02.78'	'Join-Request'		'647fda0000003fed'	
...				
'2021-09-05 11:41:31.88'	'Join-Request'		'75c600000aca2500'	
'2021-09-05 11:41:36.94'	'Join-Accept'			
'2021-09-05 11:41:39.22'	'Uplink Msg (C)'	'260b24d9'		'0000'
'2021-09-05 11:41:39.23'	'Uplink Msg (C)'	'260b24d9'		'0000'
'2021-09-05 11:42:28.75'	'Join-Request'		'647fda0000003fed'	
'2021-09-05 11:42:35.56'	'Join-Accept'			
'2021-09-05 11:42:39.47'	'Uplink Msg (C)'	'260b24d9'		'0001'
'2021-09-05 11:42:39.48'	'Uplink Msg (C)'	'260b24d9'		'0001'
'2021-09-05 11:42:51.07'	'Uplink Msg (UC)'	'260b7bbd'		'0000'
'2021-09-05 11:42:51.09'	'Uplink Msg (UC)'	'260b7bbd'		'0000'
'2021-09-05 11:43:24.07'	'Join-Request'		'75c600000aca2500'	
'2021-09-05 11:43:29.15'	'Join-Accept'			
'2021-09-05 11:43:31.39'	'Uplink Msg (C)'	'260b785b'		'0000'

FloRa + LoRaWAN Join

- Presented work to LoRa Alliance
- One of our countermeasures to be present in v1.2

5G procedures

- \exists -St-Track^{implicit,passive}
- Two types of network traffic
 1. 5G core-facing
 2. Tower-facing

TrackDev mechanised



Adding to formal-analysis

- Session-sensitive/session-insensitive similar to mechanise
- Explicit trackability simple to mechanise
- Implicit trackability difficult to mechanise

Conclusions

TrackDev

New privacy notions amenable to both practice and formal verification

New attacks on LoRa Join and 5G

Implemented in a practical toolkit LoRa's privacy attacks/checking

Solutions adopted by LoRaWAN in v1.2

Mechanised in formal verification in Tamarin

Thank you for your attention!

Questions?



March 2, 2023

Ksenia Budykho

39



TrackDev mechanised

```
/****** EXPLICITTRACKABILITYLEMMA *****/
```

```
lemma linked_deveui_devaddr_onlyonce: exists-trace
```

```
"
```

```
Ex DevEUI DevEUI2 NS JoinEUI DevNonce DevNonce2 tau_c1 tau_c2 DevAddr2 #t01 #t02 #t03
```

```
.  
DeviceJoinRequest(DevEUI, NS, JoinEUI, DevNonce, tau_c1)@ t01  
& DeviceJoinRequest(DevEUI2, NS, JoinEUI, DevNonce2, tau_c2)@ t02  
& NSReceivedReKeyInd(NS, DevEUI2, DevAddr2) @t03 //link
```

```
&#t01<#t02  
&#t02<#t03
```

```
& not(DevNonce=DevNonce2)  
& not(DevEUI=DevEUI2)
```

```
// But only 1 JoinRequest makes it
```

```
& (All #i #j . OnlyOnce('NetworkServer_Receive_JoinRequest_Forward_To_JS') @ i &  
  OnlyOnce('NetworkServer_Receive_JoinRequest_Forward_To_JS') @ j ==> #i=#j)
```

```
//and no key reveal
```

```
& not(Ex Entity Type Key #kr . KeyReveal(Entity, Type, Key) @ kr)
```

```
"
```