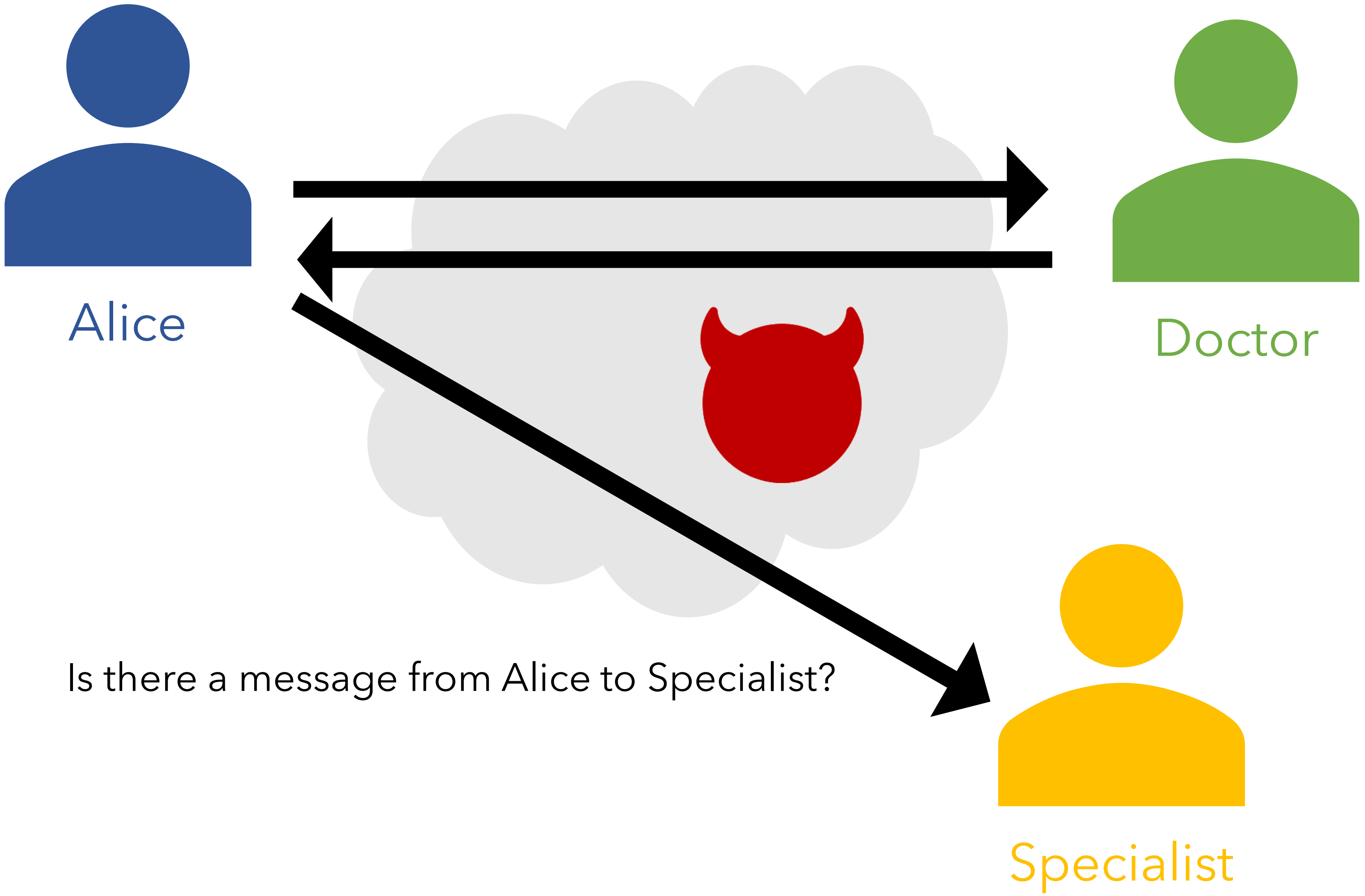
Towards Language-Based Mitigation of Traffic Analysis Attacks

Jeppe Fredsgaard Blaabjerg
jfblaa@cs.au.dk

Aslan Askarov
aslan@cs.au.dk

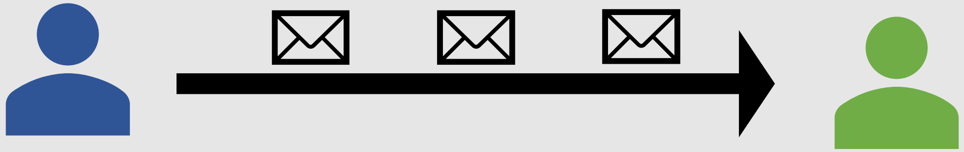
Aarhus University

Medical Referral

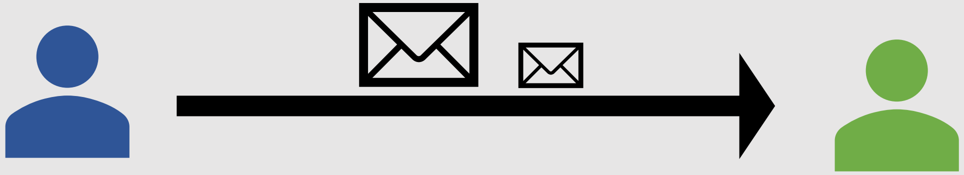


Is there a message from Alice to Specialist?

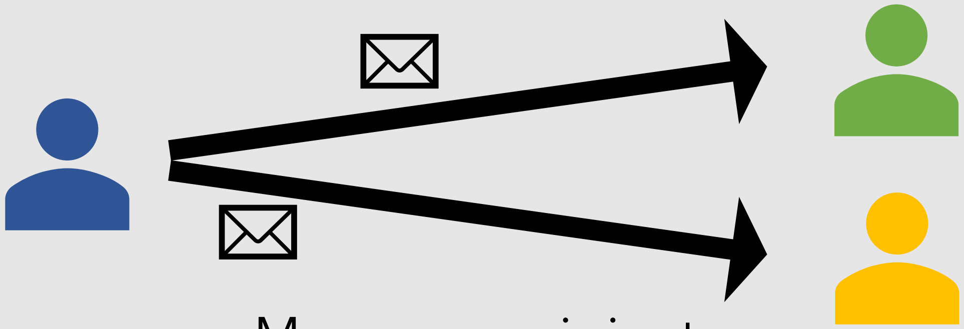
Observable Properties of Communication



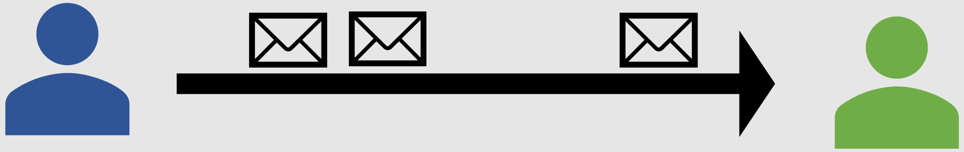
Message count



Message size

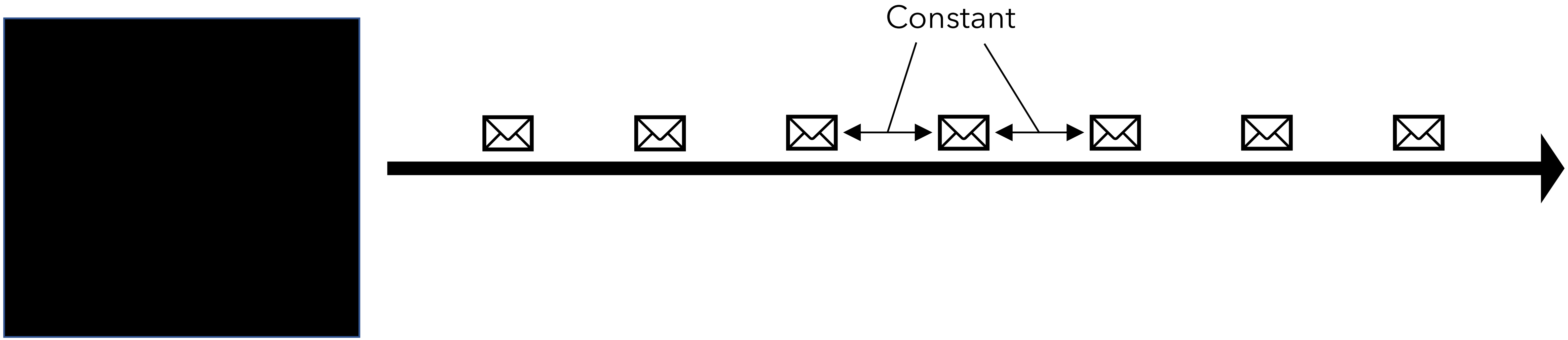


Message recipient



Message time

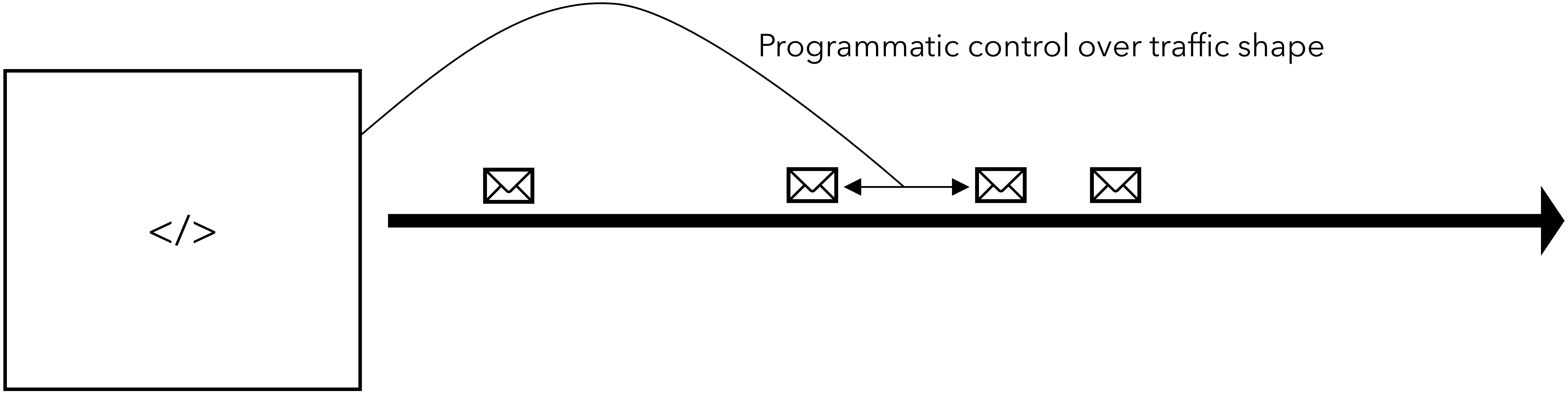
Mitigating Traffic Analysis



- System level mitigation*
  - Black-box
  - Enforcing constant rate/padding on all communication

* X. Fu, B. Graham, R. Bettati, W. Zhao, and D. Xuan, “Analytical and empirical analysis of countermeasures to traffic analysis attacks,” 2003 International Conference on Parallel Processing,
K. P. Dyer, S. E. Coull, T. Ristenpart, and T. Shrimpton, “Peek-a-boo, I still see you: Why efficient traffic analysis countermeasures fail,” 2012 IEEE Symposium on Security and Privacy.

Our Approach



- Language-based mitigation
  - Program source to control traffic shape
  - Information flow control to enforce that traffic shape does not leak secrets
- Fixed-size packets
  - Messages encoded as sequences of packets
  - Packet contents protected by encryption

SELENE: Language and Runtime

- Simple imperative language with I/O
- Labelled channels – assuming a single channel per level ℓ
- Runtime support for scheduling and queueing messages

$e ::= n \mid x \mid e \text{ op } e$

$c ::= x = e \mid c_1;\ c_2 \mid \text{skip} \mid \text{if } e \text{ then } c_1 \text{ else } c_2 \mid \text{while } e \text{ do } c \mid \text{sleep } (e)$

$\quad \mid x = \text{input } (\ell)$ (* input from channel ℓ *)

$\quad \mid x = \text{sizeof } (e)$ (* compute runtime size of an expression *)

$\quad \mid \text{schedule } (\ell, e_1, e_2)$ (* schedule packets on channel ℓ *)

$\quad \mid \text{queue } (\ell, e)$ (* queue message on channel ℓ *)

queue (ℓ, *v*) places value *v* into the output buffer for channel ℓ
- splits the value into packets as necessary

schedule (ℓ, *n*, *t*) allocates *n* packets to be sent on channel ℓ starting after time delay *t*

Schedule/Queue semantics

- Corner case 1
  - Packets scheduled but nothing queued – send dummy packets
- Corner case 2
  - Packets queued but not scheduled – buffer until schedule is set
- Schedule is globally deterministic
  - The semantics keeps track of time counter *t*

schedule config at *t*=99

Time	100	101	102	103	104	105	106
Channel		Alice	Alice		Bob		

instruction at t=100

schedule (Charlie, 4, 0)

Schedule four packets for Charlie with no time delay

schedule config at *t*=100

Time	100	101	102	103	104	105	106
Channel	**Charlie**	Alice	Alice	**Charlie**	Bob	**Charlie**	**Charlie**

Medical Referral Revisited

```
size_result = sizeof(result);
schedule(Doctor,size_result,5);

size_id = sizeof(id);
schedule(Specialist,size_id,100);

queue(Doctor,result);

needs_treatment = input(Doctor);

if needs_treatment
then queue(Specialist,id);
else skip;
```

Type System

T-If

$$\frac{\Gamma \vdash e : \mathbf{int}@\ell \qquad \Gamma, pc \sqcup \ell \vdash c_1 : pc' \qquad \Gamma, pc \sqcup \ell \vdash c_2 : pc''}{\Gamma, pc \vdash \texttt{if}\ \ e\ \ \texttt{then}\ \ c_1\ \ \texttt{else}\ \ c_2 : pc' \sqcup pc''}$$

T-Schedule

$$\frac{pc = \bot \qquad \Gamma \vdash e_1 : \mathbf{int}@\bot \qquad \Gamma \vdash e_2 : \mathbf{int}@\bot}{\Gamma, pc \vdash \texttt{schedule}(\ell, e_1, e_2) : pc}$$

T-Queue

$$\frac{\Gamma \vdash e : \sigma_e@\ell_e \qquad \ell_e \sqcup pc \sqsubseteq \ell}{\Gamma, pc \vdash \texttt{queue}(\ell, e) : pc}$$

Medical Referral Re-revisited

```
size_result = sizeof(result);
schedule(Doctor,size_result,5);

queue(Doctor,result);

needs_treatment = input(Doctor);

if needs_treatment
then {
  size_id = sizeof(id);
  schedule(Specialist,size_id,5);
  queue(Specialist,id);
}
else skip;
```

Security Condition

$$k(\mathsf{cfg}, \tau, \ell\,) = \{\mathsf{cfg}' | \mathsf{cfg} \approx_{\ell} \mathsf{cfg}' \land \mathsf{cfg}' \rightarrow^{*}_{\tau'} \mathsf{cfg}'' \land \tau \approx_{\ell} \tau'\}$$

Attacker knowledge[1]

$$k(\mathsf{cfg}, \tau \cdot \alpha, \ell) \supseteq k(\mathsf{cfg}, \tau, \ell)$$

Security condition

Soundness theorem

- Well-typed SELENE programs do not leak by their output

[1] Askarov and A. Sabelfeld, "Gradual release: Unifying declassification, encryption and key release policies," 2007 IEEE Symposium on Security and Privacy.

SELENE Limitations

- Scheduling requires low pc
  - Progress-sensitive type system leads to pc-creep
- Possible approaches for mitigating pc-creep
  - Extra precision in the static reasoning
  - Declassification
    - Value declassification (e.g., before branching)
    - PC-declassification[2]

[2] J. Bay and A. Askarov, "Reconciling progress-insensitive noninterference and declassification," CSF 2020

Towards Language-Based Mitigation of Traffic Analysis Attacks

/ Takeaways

- Traffic analysis is a significant concern
    - IFC models should reflect that
- Language-based solutions
    - Potential to reduce overhead compared to black-box techniques
- Future research
    - New language designs for mitigating traffic analysis

Thank you!

jfblaa@cs.au.dk