# No Right to Remain Silent: Isolating Malicious Mixes

Hemi Leibowitz[1] Ania M. Piotrowska[2] George Danezis[2] Amir Herzberg[3]

[1]Bar-Ilan University, IL [2]University College London, UK [3]University of Connecticut, US

# INTRODUCTION (what the paper is all about)

- Anonymity is important and challenging

# INTRODUCTION (what the paper is all about)

- Anonymity is important and challenging

| System | Efficiency | Security |
|---|---|---|
| **Onion routing (e.g., Tor)** | Efficient, low-latency, practical, popular | |

**“Anonymity loves company”**

# INTRODUCTION (what the paper is all about)

- Anonymity is important and challenging

| System | Efficiency | Security |
|---|---|---|
| **Onion routing (e.g., Tor)** | Efficient, low-latency, practical, popular | Insecure against timing attacks |

**From "Tor: The Second-Generation Onion Router":**

"Tor does not claim to completely solve end-to-end timing or intersection attacks."

# INTRODUCTION (what the paper is all about)

- Anonymity is important and challenging

| System | Efficiency | Security |
|---|---|---|
| **Onion routing (e.g., Tor)** | Efficient, low-latency, practical, popular | Insecure against timing attacks |
| **Classic mixnets** | Efficient, higher latency | Secure against global eavesdropper and curious not malicious servers (mixes) |

# INTRODUCTION (what the paper is all about)

- Anonymity is important and challenging

| System | Efficiency | Security |
|---|---|---|
| **Onion routing (e.g., Tor)** | Efficient, low-latency, practical, popular | Insecure against timing attacks |
| **Classic mixnets** | Efficient, higher latency | Secure against global eavesdropper and curious not malicious servers (mixes) |
| **Dining Cryptographers networks (DCnets), secure shuffle** | **High overhead** (computing and/or communication) | Secure against global eavesdropper and **malicious servers** |

# INTRODUCTION (what the paper is all about)

- Anonymity is important and challenging

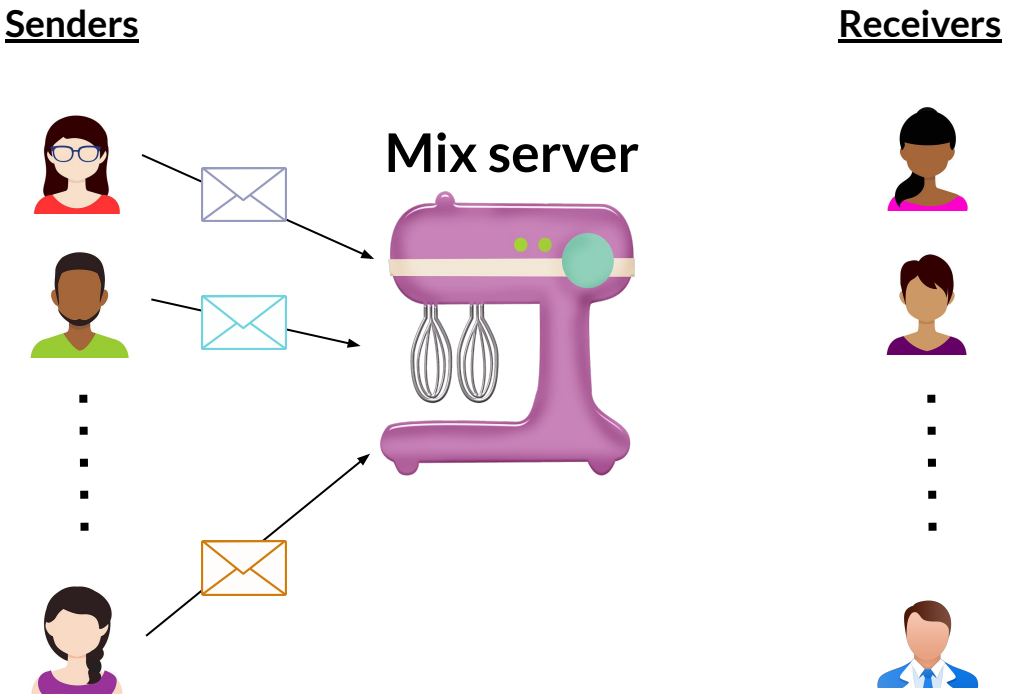
| System | Efficiency | Security |
|---|---|---|
| **Onion routing (e.g., Tor)** | Efficient, low-latency, practical, popular | Insecure against timing attacks |
| **Classic mixnets** | Efficient, higher latency | Secure against global eavesdropper and curious not malicious servers (mixes) |
| **Miranda's mixnet (this work)** | Efficient, higher latency | Secure against global eavesdropper and **malicious servers (mixes)** |
| **Dining Cryptographers networks (DCnets), secure shuffle** | **High overhead** (computing and/or communication) | Secure against global eavesdropper and **malicious servers** |

# INTRODUCTION (basic concepts: mix servers, mix cascades and mix net)

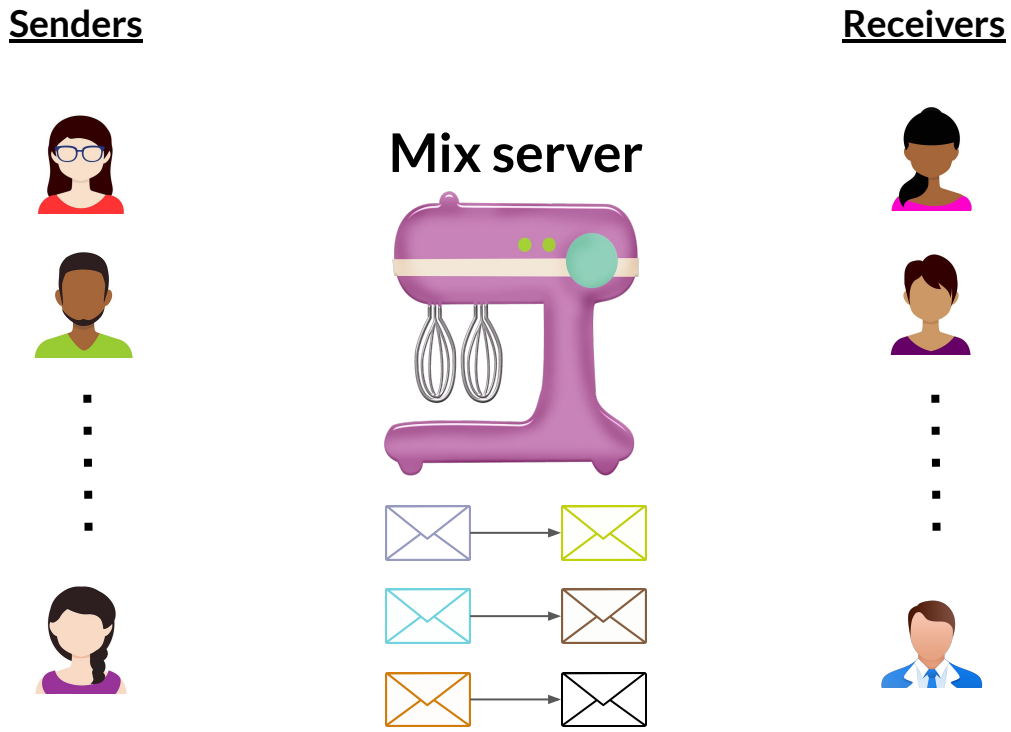
**Senders**

**Receivers**

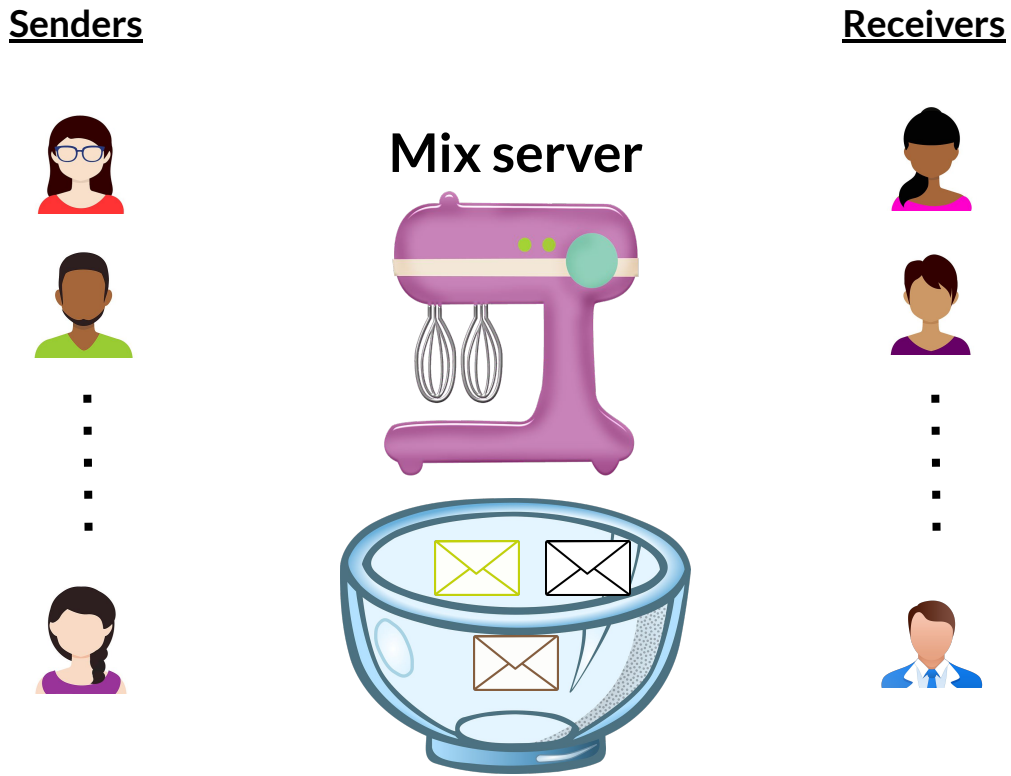
# INTRODUCTION (basic concepts: mix servers, mix cascades and mix net)



# INTRODUCTION (basic concepts: mix servers, mix cascades and mix net)

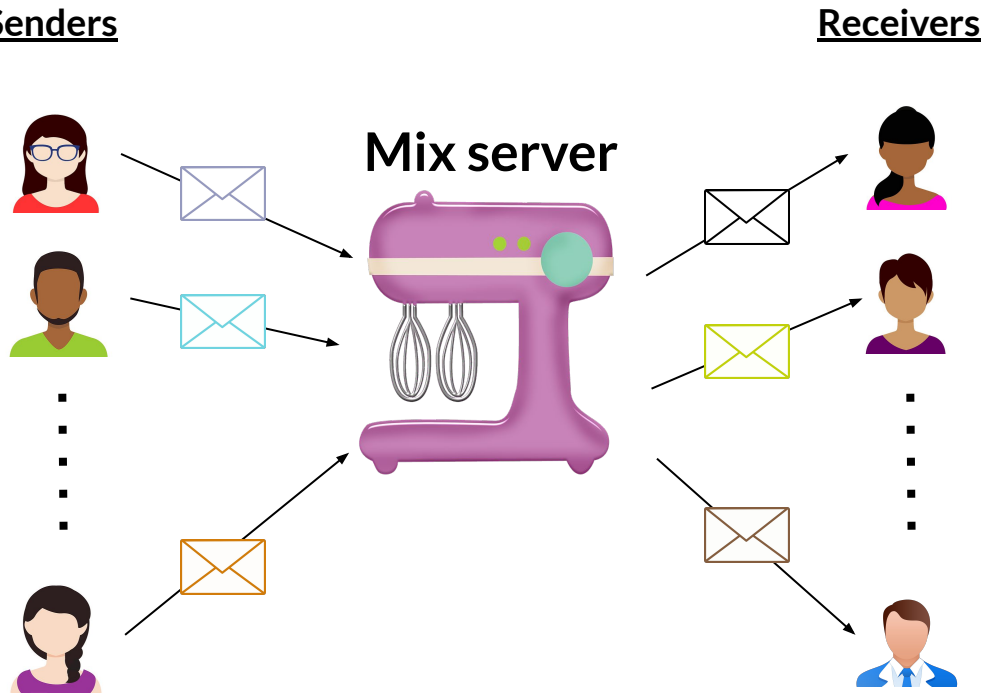


# INTRODUCTION (basic concepts: mix servers, mix cascades and mix net)



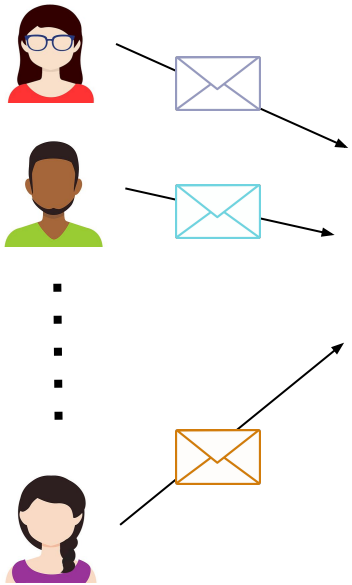
# INTRODUCTION (basic concepts: mix servers, mix cascades and mix net)

**Incoming and outgoing messages are unlinkable,**
**to an outside observer**



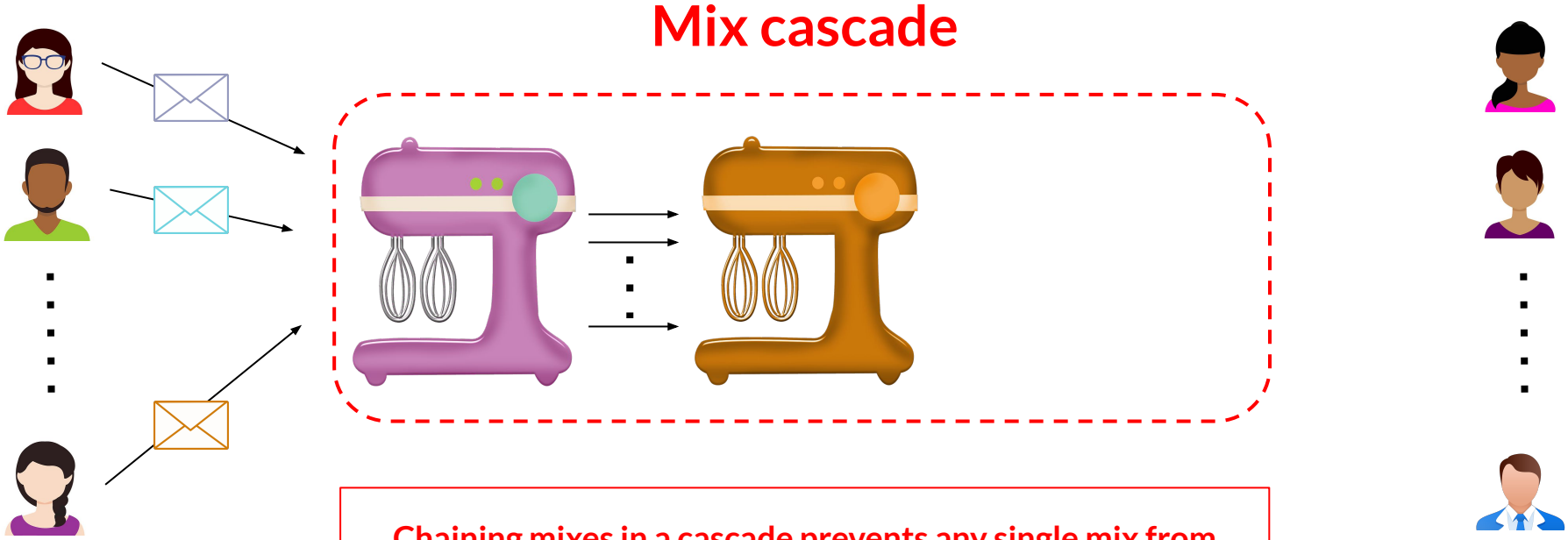
# INTRODUCTION (basic concepts: mix servers, mix cascades and mix net)

## Mix cascade



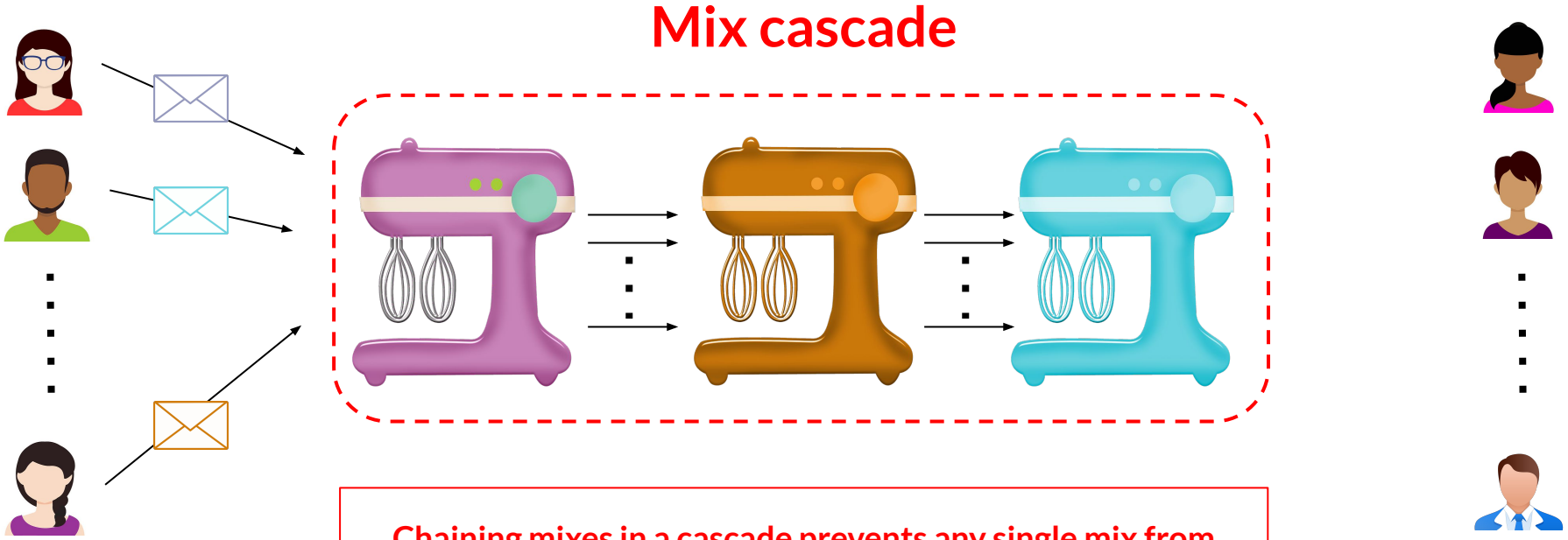
**Chaining mixes in a cascade prevents any single mix from tracking messages end-to-end.**

# INTRODUCTION (basic concepts: mix servers, mix cascades and mix net)



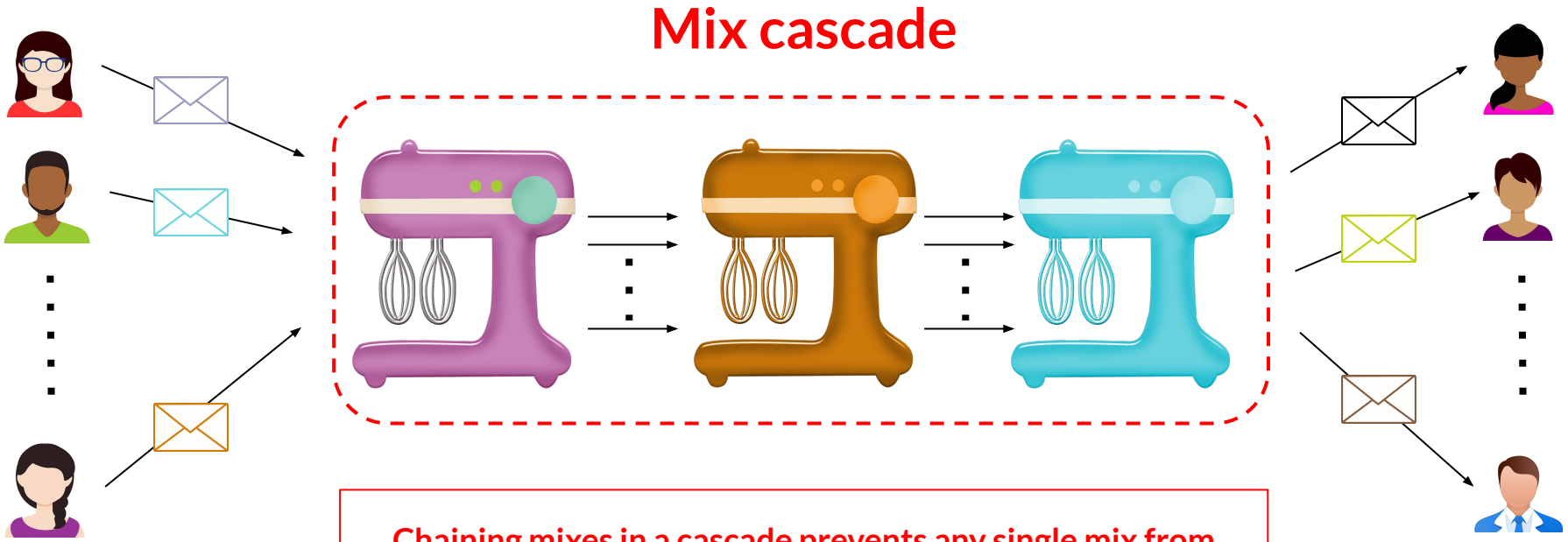

Chaining mixes in a cascade prevents any single mix from tracking messages end-to-end.

# INTRODUCTION (basic concepts: mix servers, mix cascades and mix net)



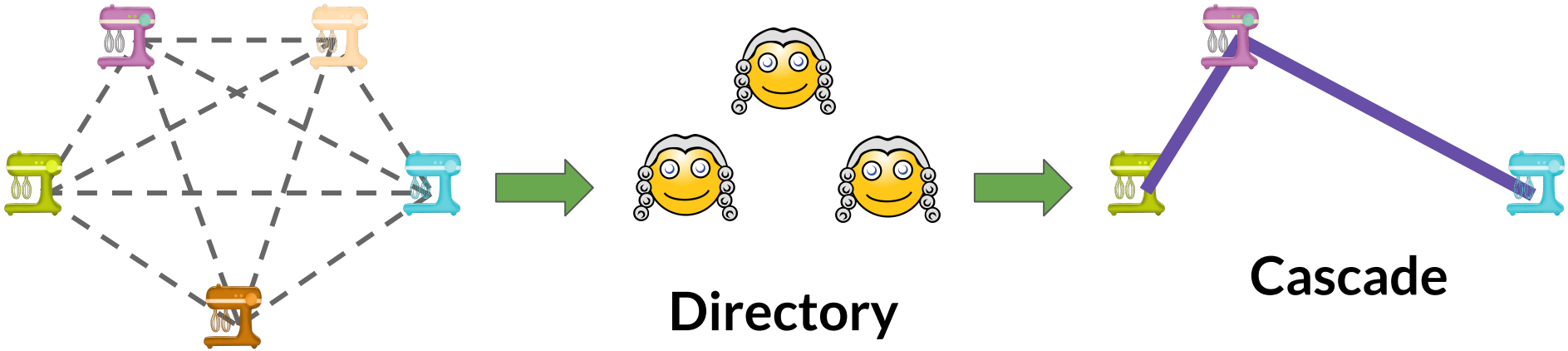

Chaining mixes in a cascade prevents any single mix from tracking messages end-to-end.

# INTRODUCTION (basic concepts: mix servers, mix cascades and mix net)

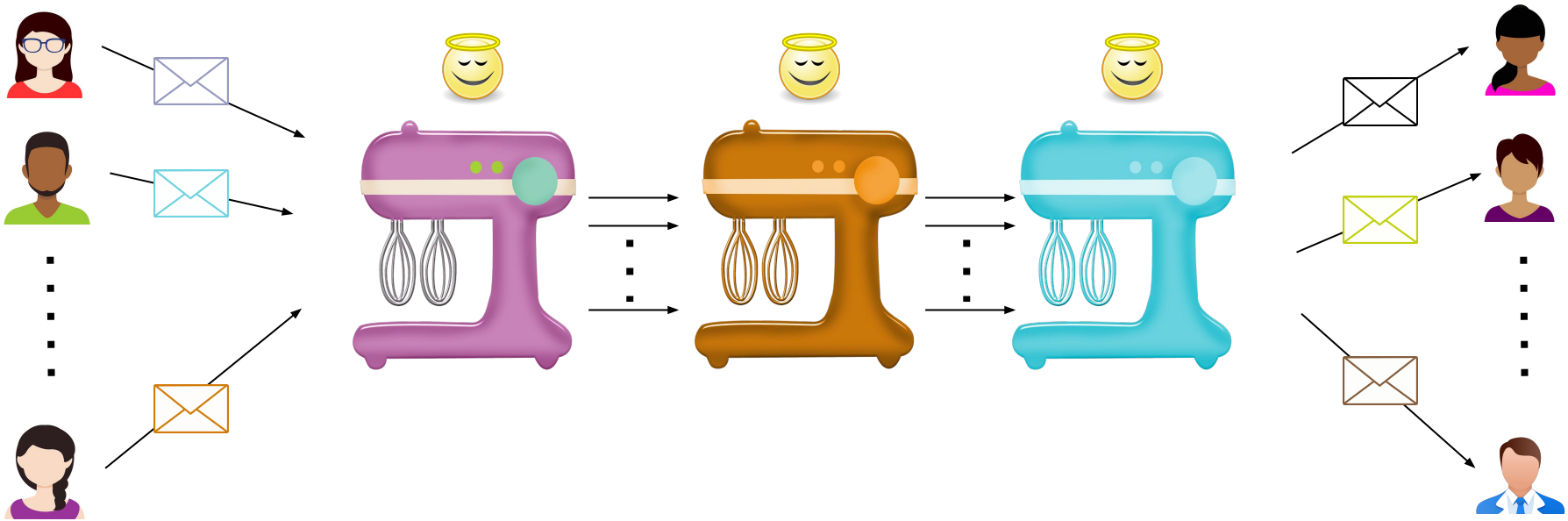



Chaining mixes in a cascade prevents any single mix from tracking messages end-to-end.

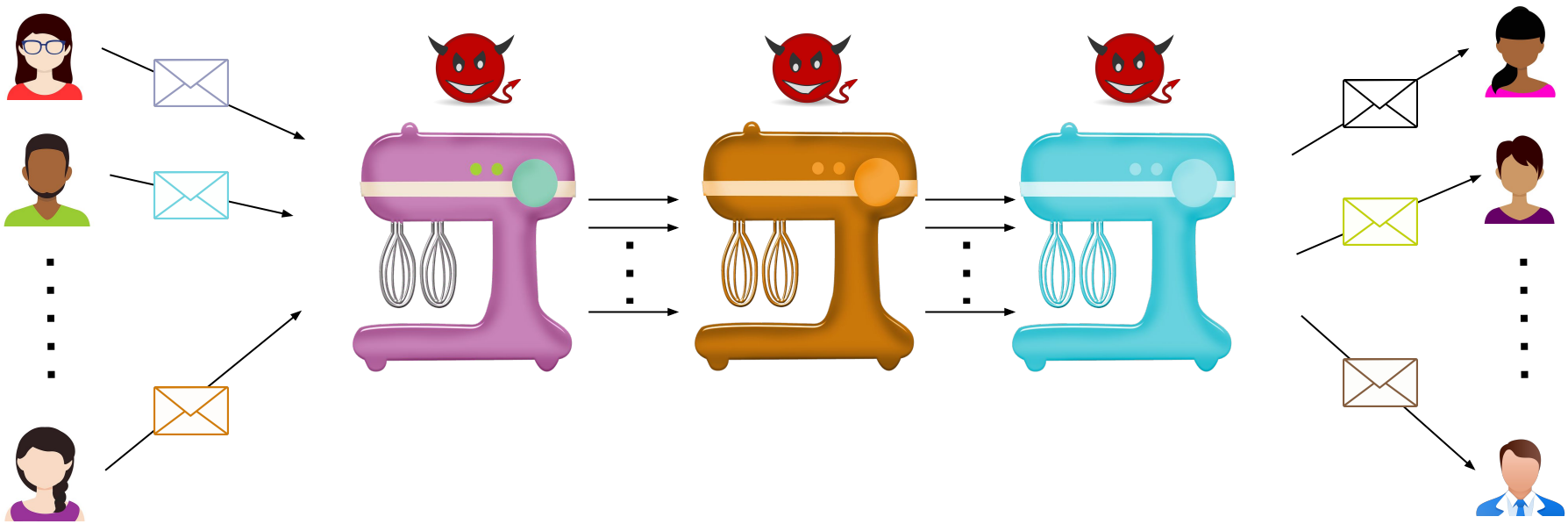
# INTRODUCTION (basic concepts: mix servers, mix cascades and mix net)



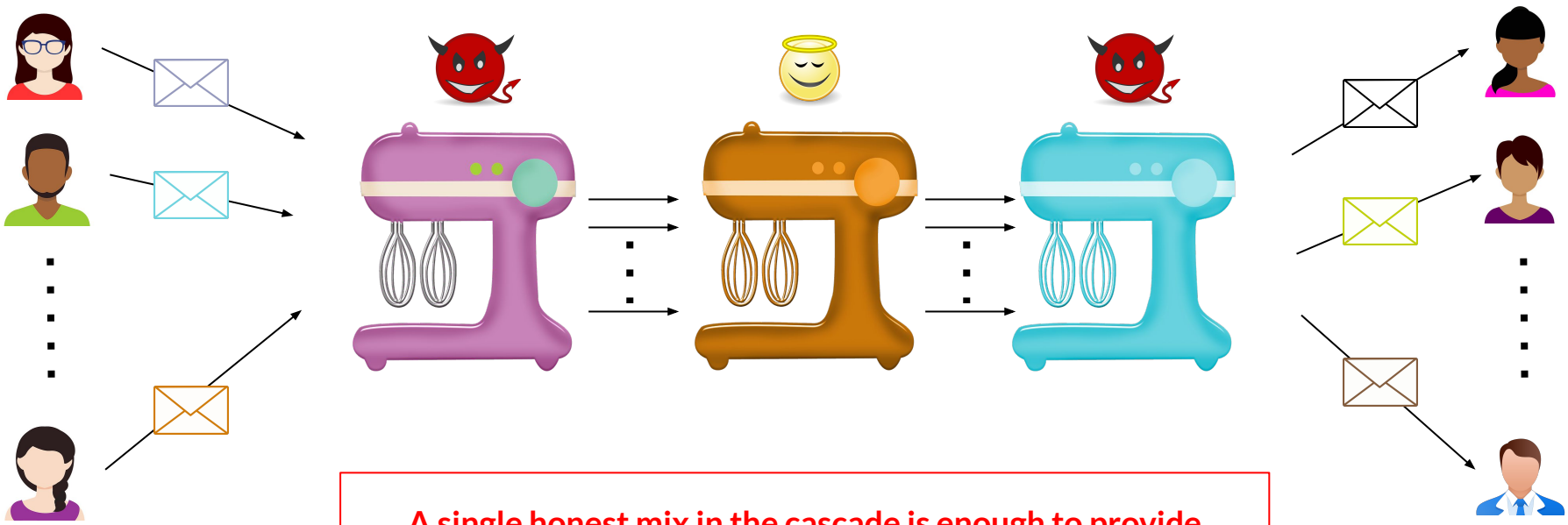
# INTRODUCTION (basic concepts: mix servers, mix cascades and mix net)



# INTRODUCTION (basic concepts: mix servers, mix cascades and mix net)

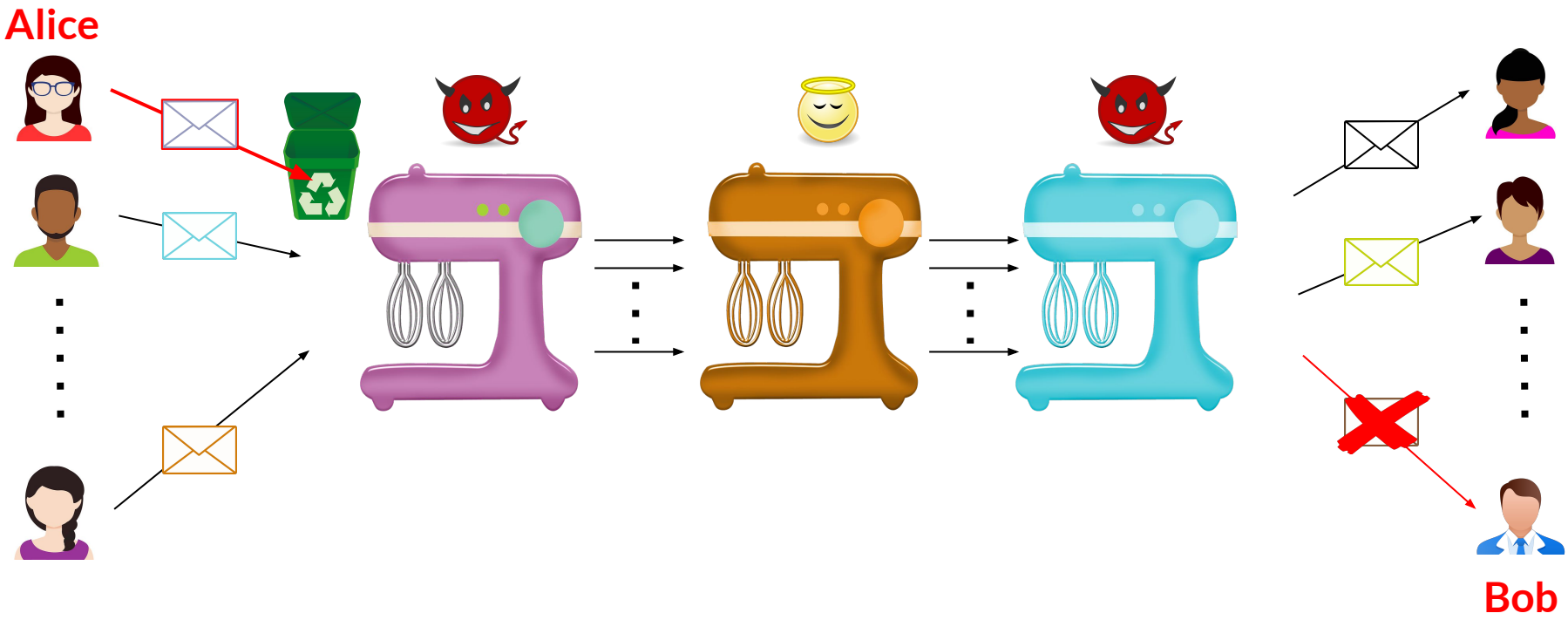


# INTRODUCTION (basic concepts: mix servers, mix cascades and mix net)

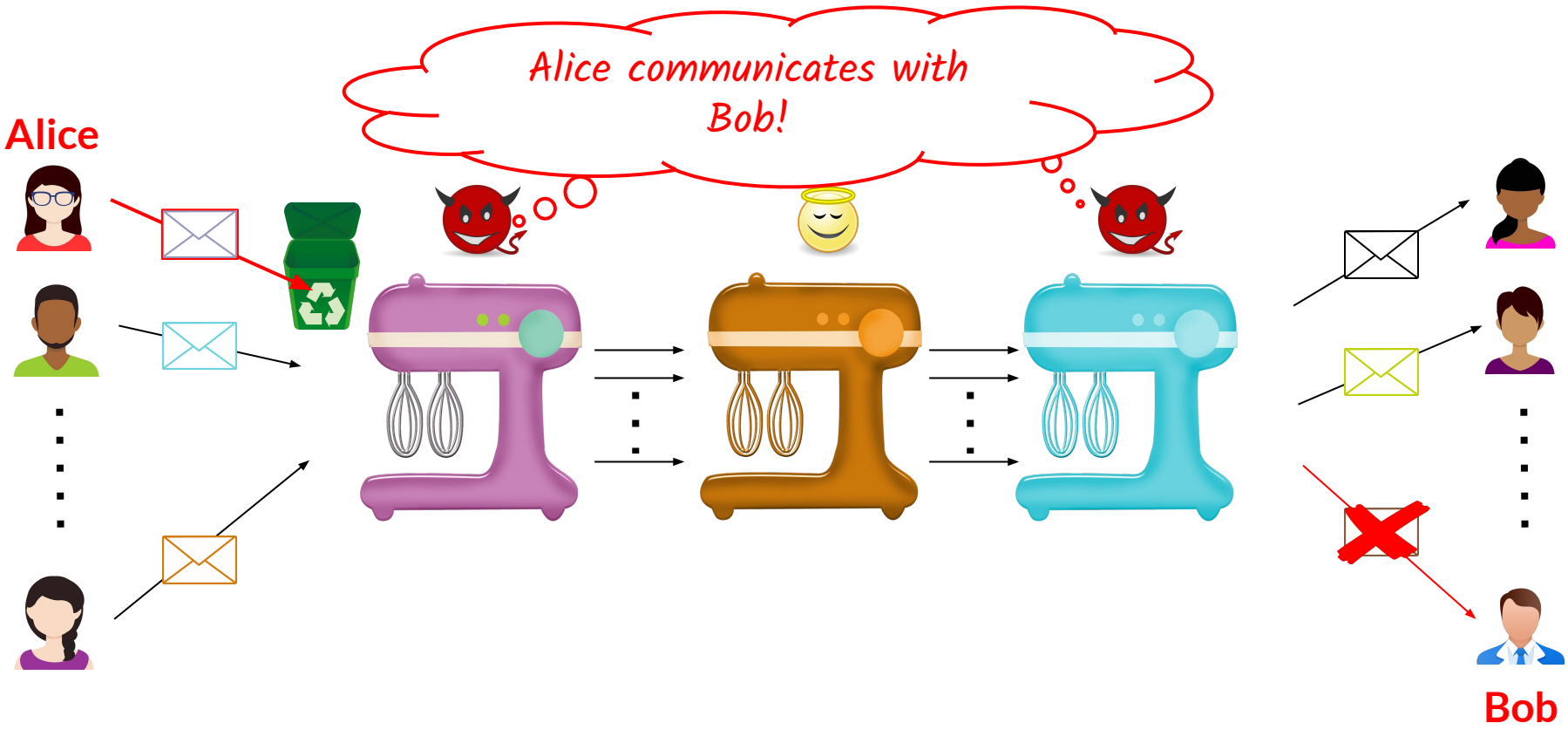


A single honest mix in the cascade is enough to provide protection against passive global observer

# MOTIVATION (why malicious mixes are a threat to mixnets)



# MOTIVATION (why malicious mixes are a threat to mixnets)



# MIRANDA'S DESIGN (assumptions)

- A fixed set of mixes (no churn)
- More honest mixes than malicious mixes (no Sybil)
- Reliable communication and processing
- Synchronized clocks

# MIRANDA'S DESIGN (challenges)

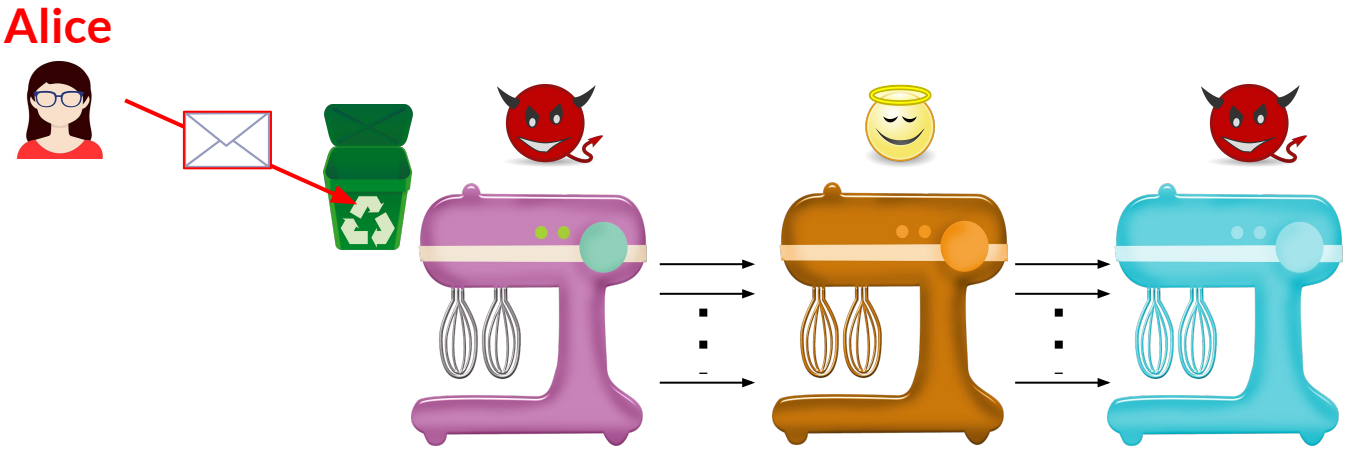
- Detect attacks by malicious mixes
- Penalize the malicious mix
- Identify the malicious mix

# MIRANDA'S DESIGN (challenges)

✔ Detect attacks by malicious mixes

- **Penalize the malicious mix**
- Identify the malicious mix

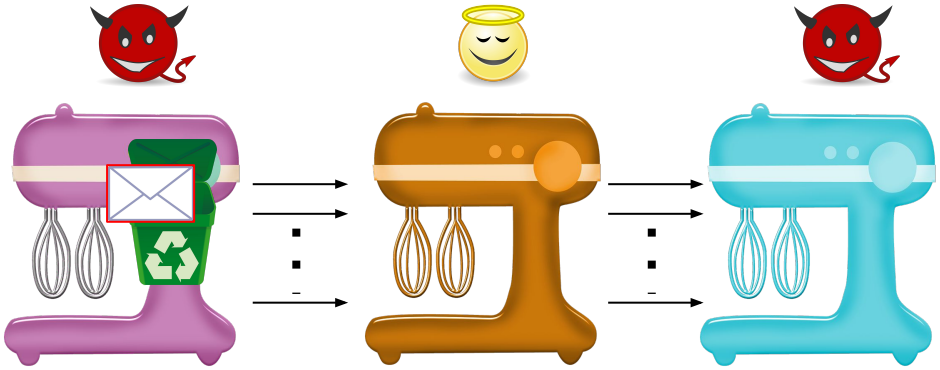
# IDENTIFYING THE MALICIOUS MIX IS CHALLENGING



# IDENTIFYING THE MALICIOUS MIX IS CHALLENGING

**Alice**

My message never arrived to its destination. Did you forward it?!

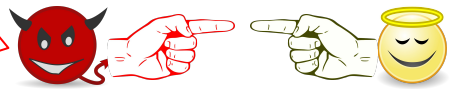


# IDENTIFYING THE MALICIOUS MIX IS CHALLENGING

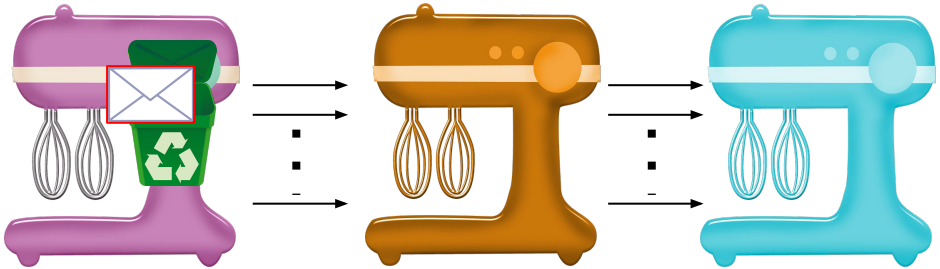
**Alice**

My message never arrived to its destination. Did you forward it?!

Of course I've forwarded it!



Hey!!! I didn't get this message!

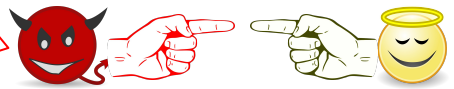


# IDENTIFYING THE MALICIOUS MIX IS CHALLENGING

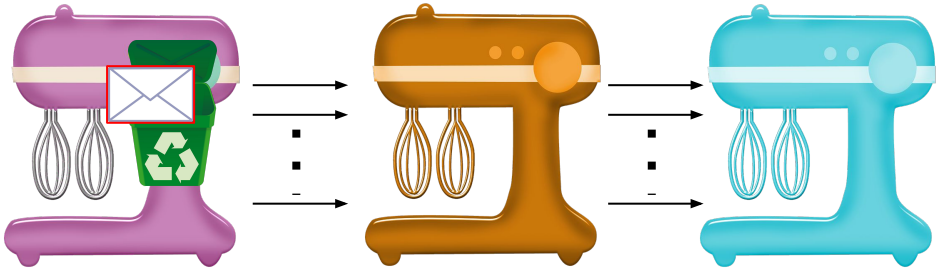
**Alice**

My message never arrived to its destination. Did you forward it?!

Of course I've forwarded it!



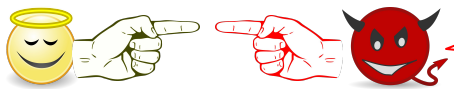
Hey!!! I didn't get this message!



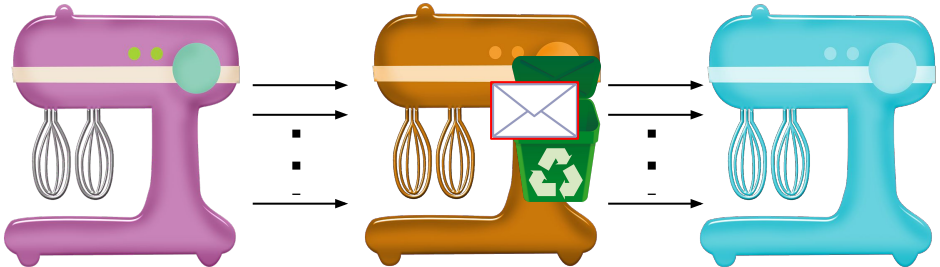
**Alice**

My message never arrived to its destination. Did you forward it?!

Of course I've forwarded it!



Hey!!! I didn't get this message!

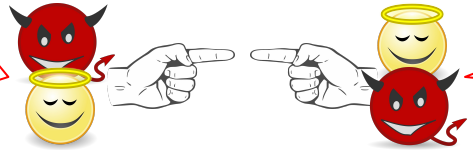


# MIRANDA *(focus on problematic pair of mixes)*

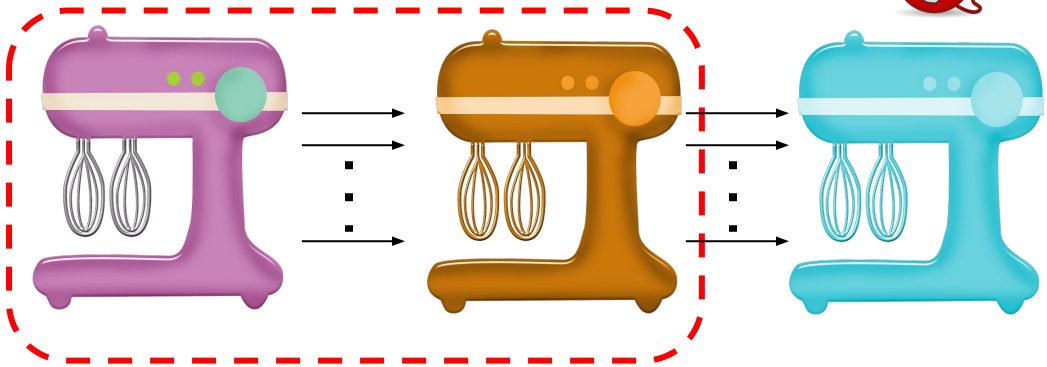
**Alice**

My message never arrived to its destination. Did you forward it?!

Of course I've forwarded it!



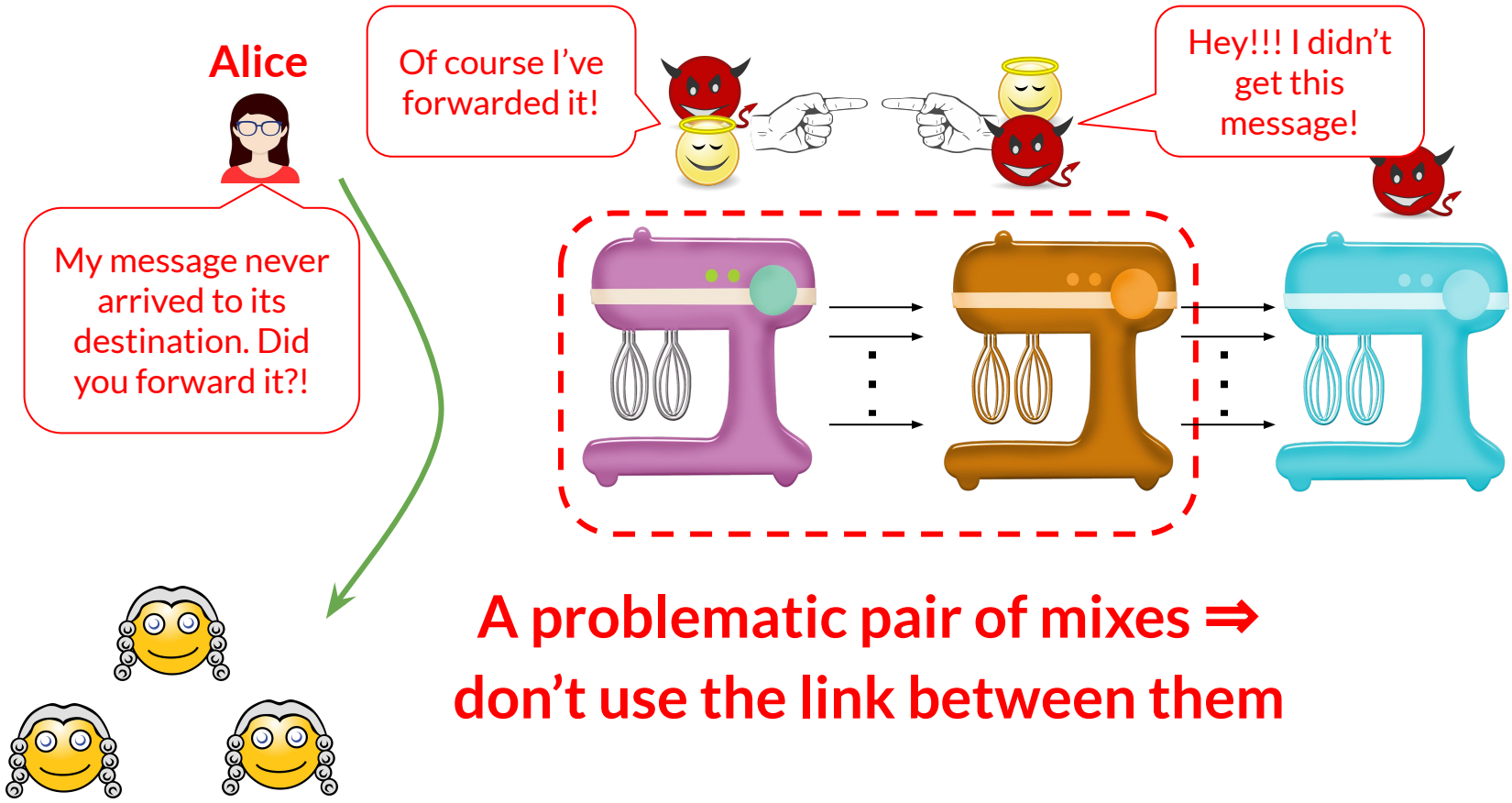
Hey!!! I didn't get this message!



**A problematic pair of mixes ⇒**

**don't use the link between them**

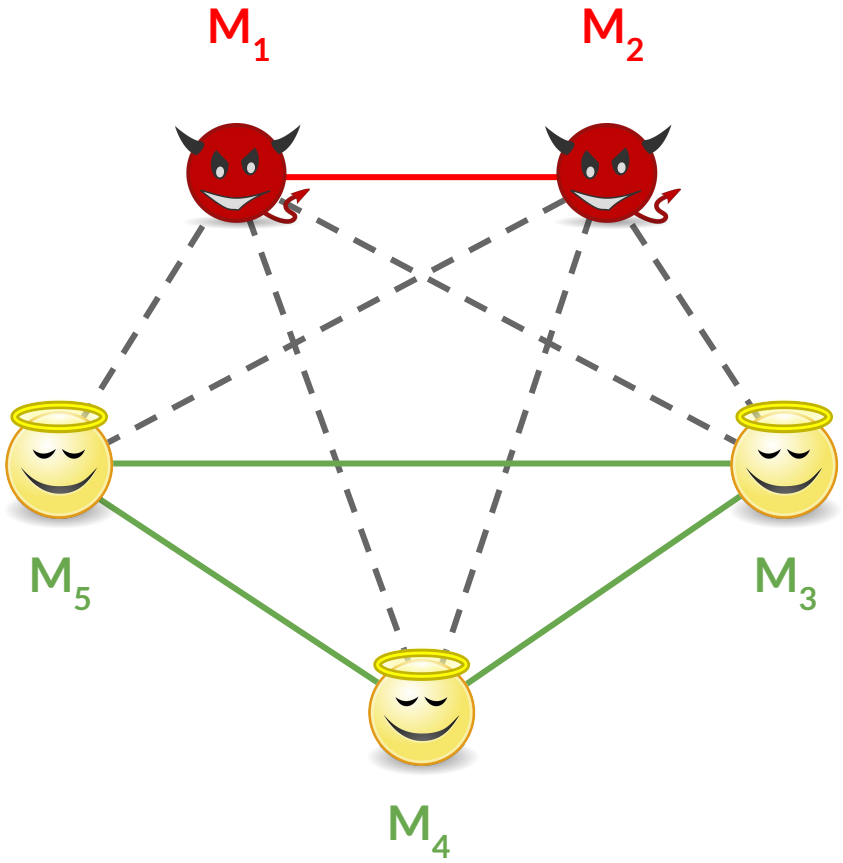
# MIRANDA (focus on problematic pair of mixes)




**A problematic pair of mixes ⇒ don't use the link between them**

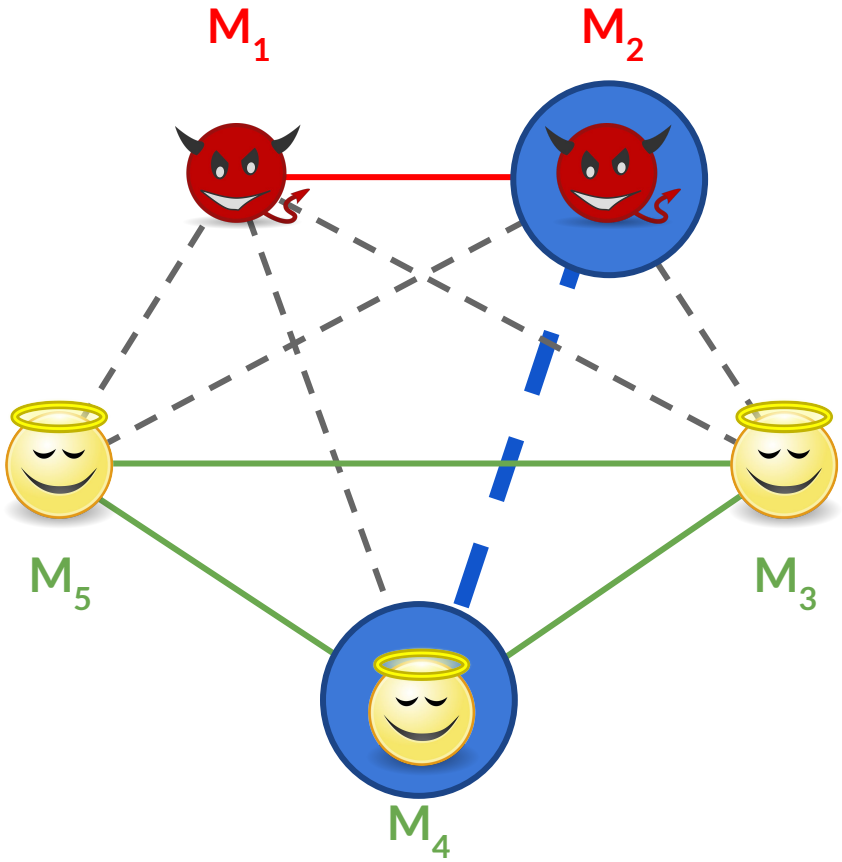
# MIRANDA FROM 10,000 FEET (removing problematic links is a good idea)

In the beginning,
everyone are willing to communicate with each other



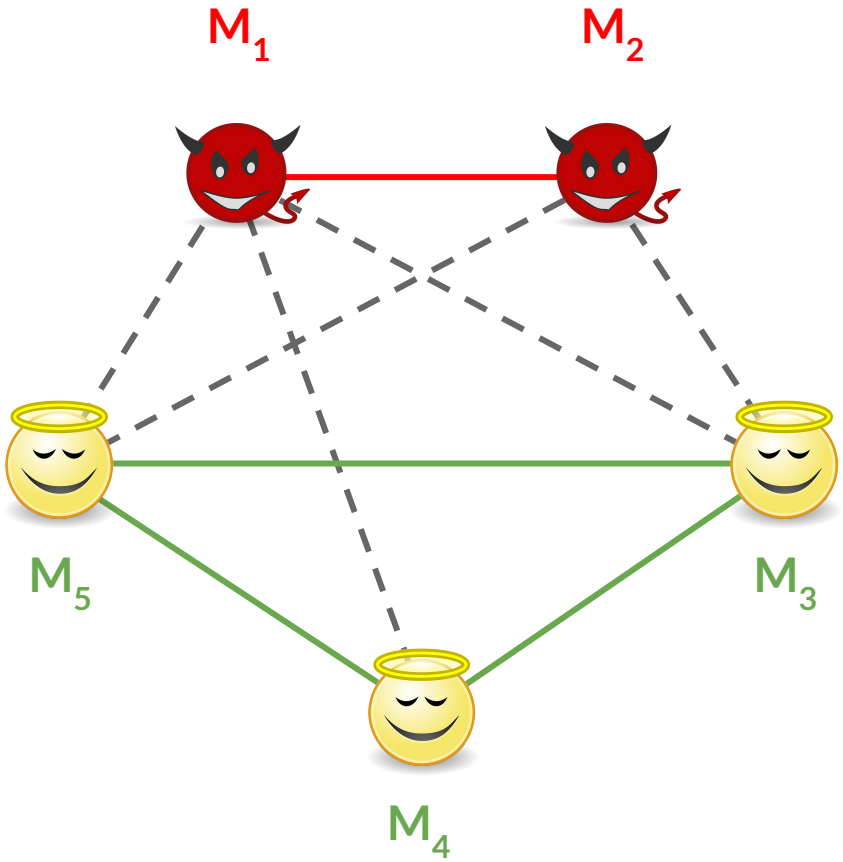
# MIRANDA FROM 10,000 FEET (removing problematic links is a good idea)

Miranda detects problematic pairs involved in active attacks, and removes their shared links, eliminating these attack vectors



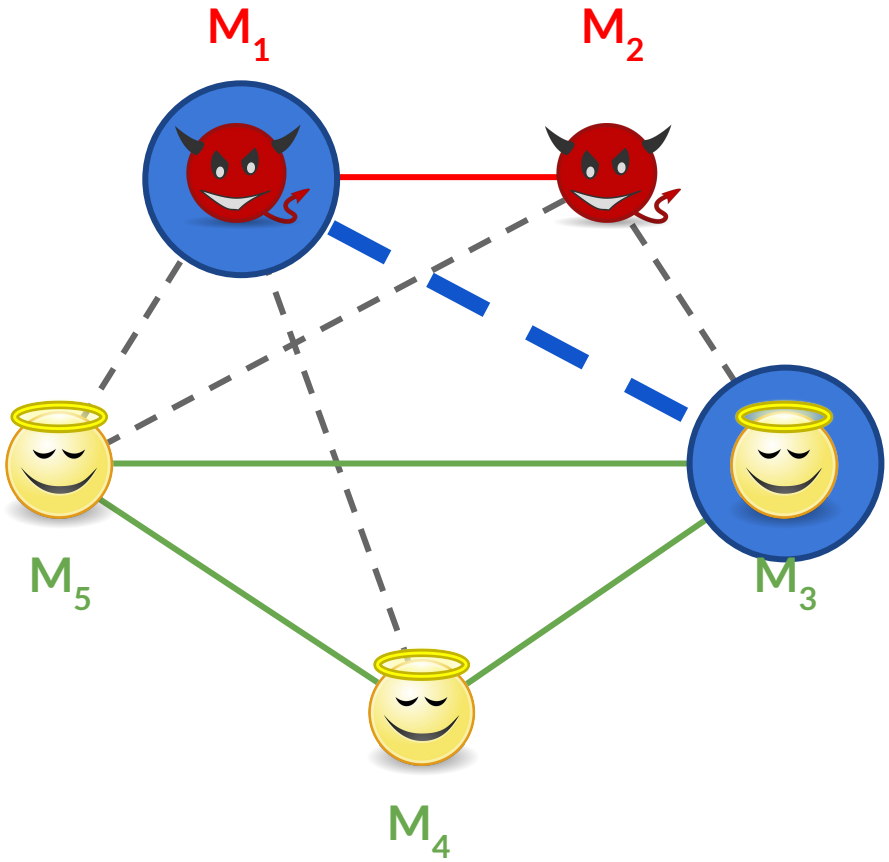
# MIRANDA FROM 10,000 FEET (removing problematic links is a good idea)

Miranda detects problematic pairs involved in active attacks, and removes their shared links, eliminating these attack vectors



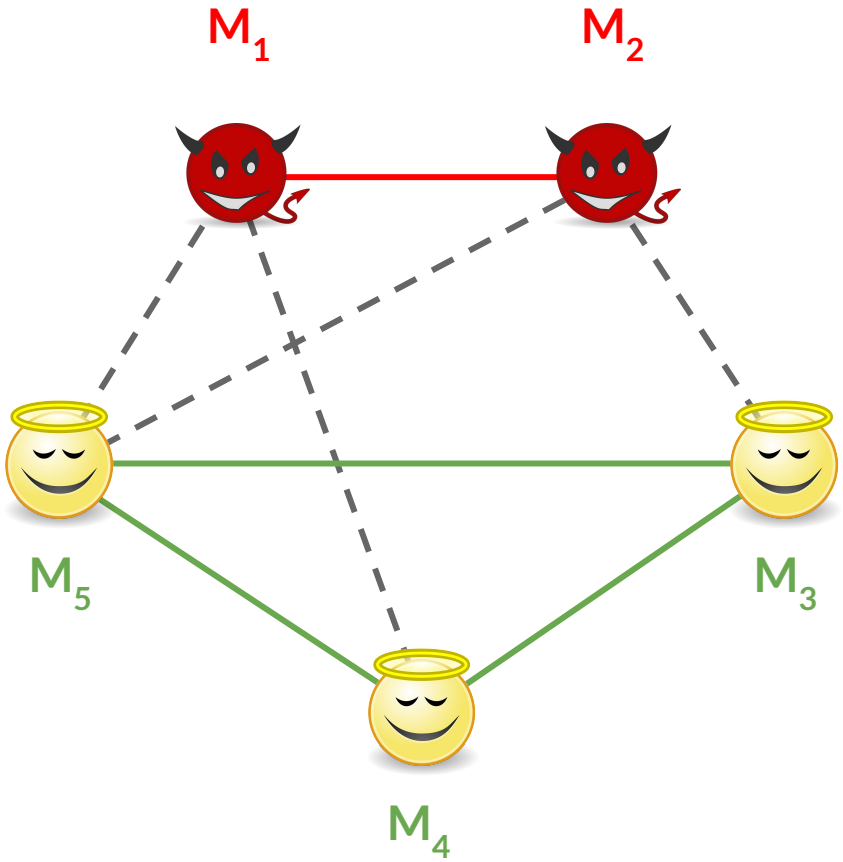
# MIRANDA FROM 10,000 FEET (removing problematic links is a good idea)

Miranda detects problematic pairs involved in active attacks, and removes their shared links, eliminating these attack vectors



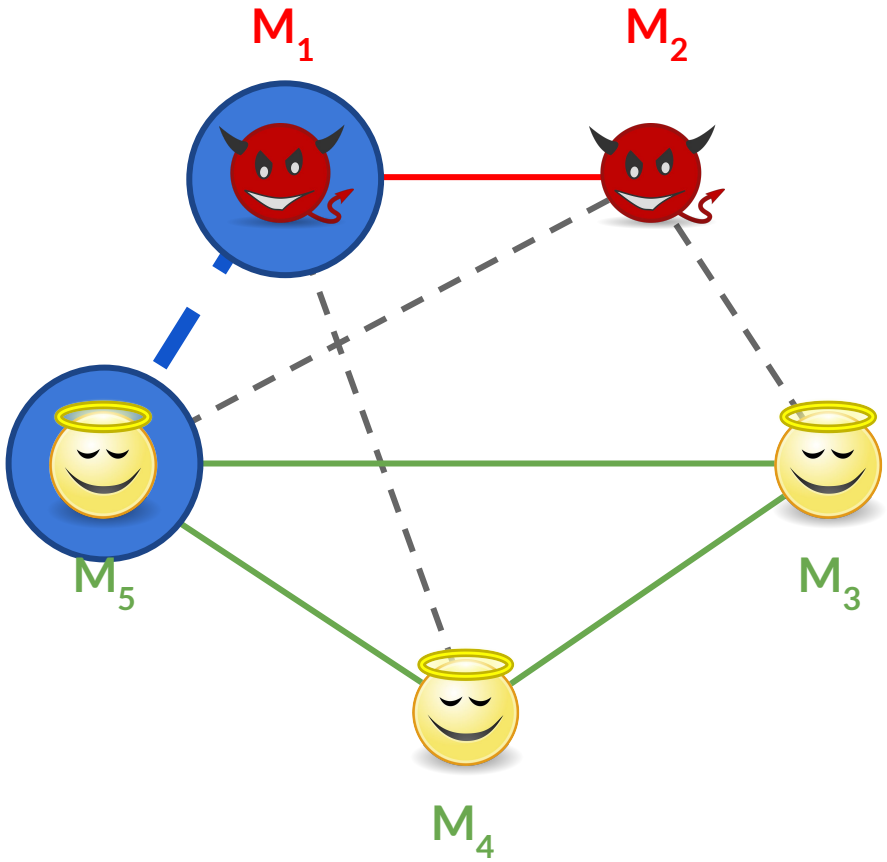
# MIRANDA FROM 10,000 FEET (removing problematic links is a good idea)

Miranda detects problematic pairs involved in active attacks, and removes their shared links, eliminating these attack vectors



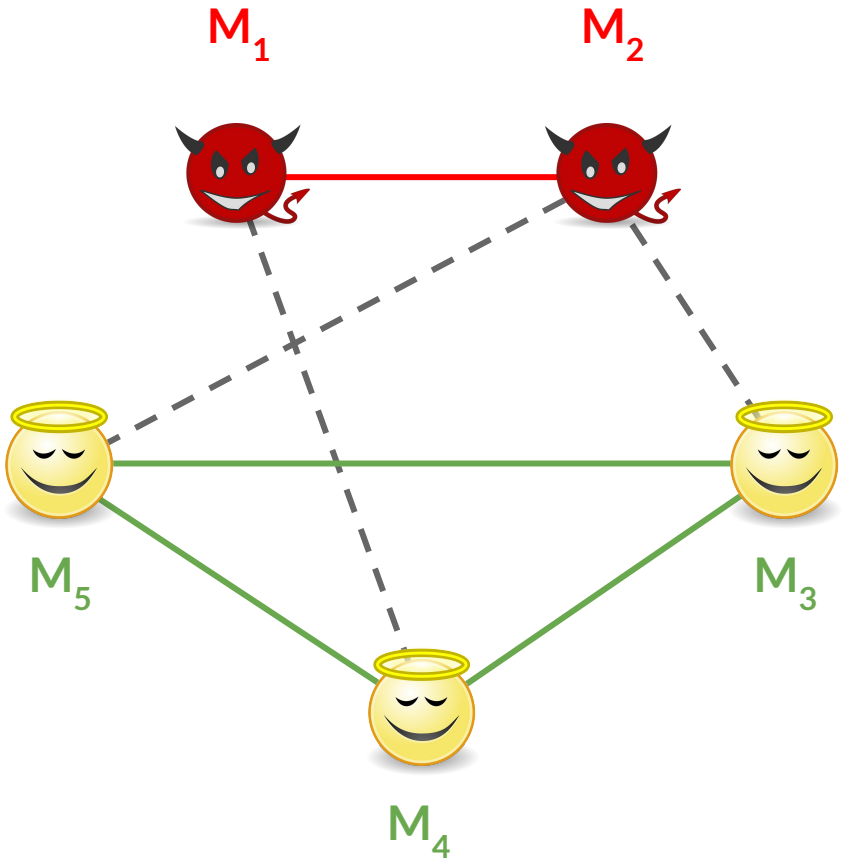
# MIRANDA FROM 10,000 FEET (removing problematic links is a good idea)

Miranda detects problematic pairs involved in active attacks, and removes their shared links, eliminating these attack vectors



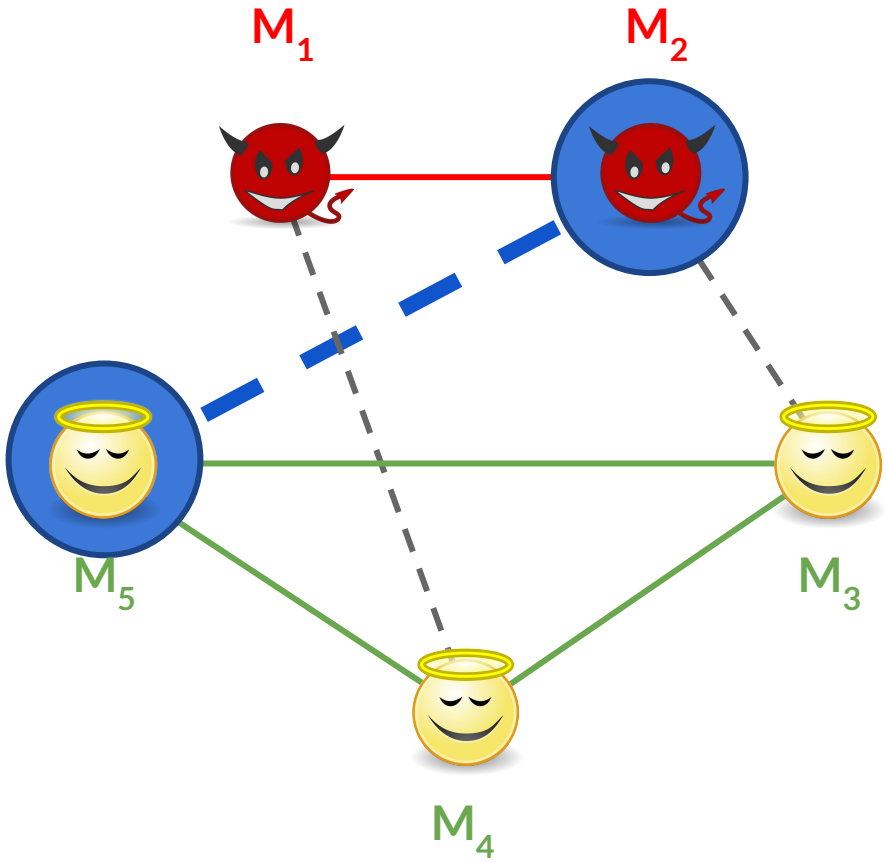
# MIRANDA FROM 10,000 FEET (removing problematic links is a good idea)

Miranda detects problematic pairs involved in active attacks, and removes their shared links, eliminating these attack vectors



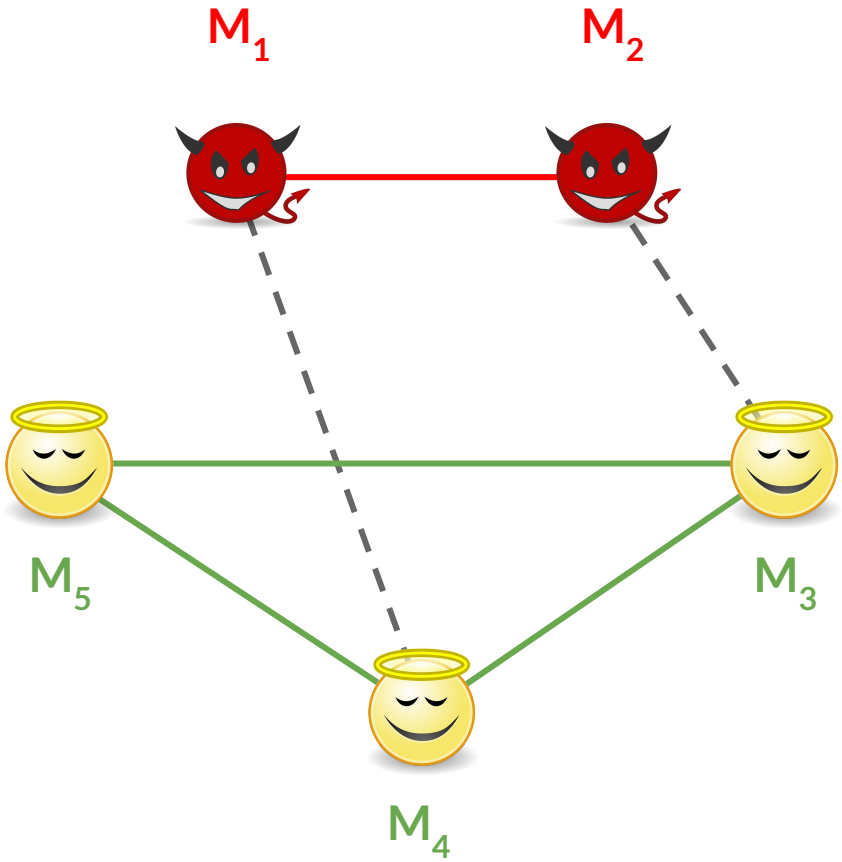
# MIRANDA FROM 10,000 FEET (removing problematic links is a good idea)

Miranda detects problematic pairs involved in active attacks, and removes their shared links, eliminating these attack vectors



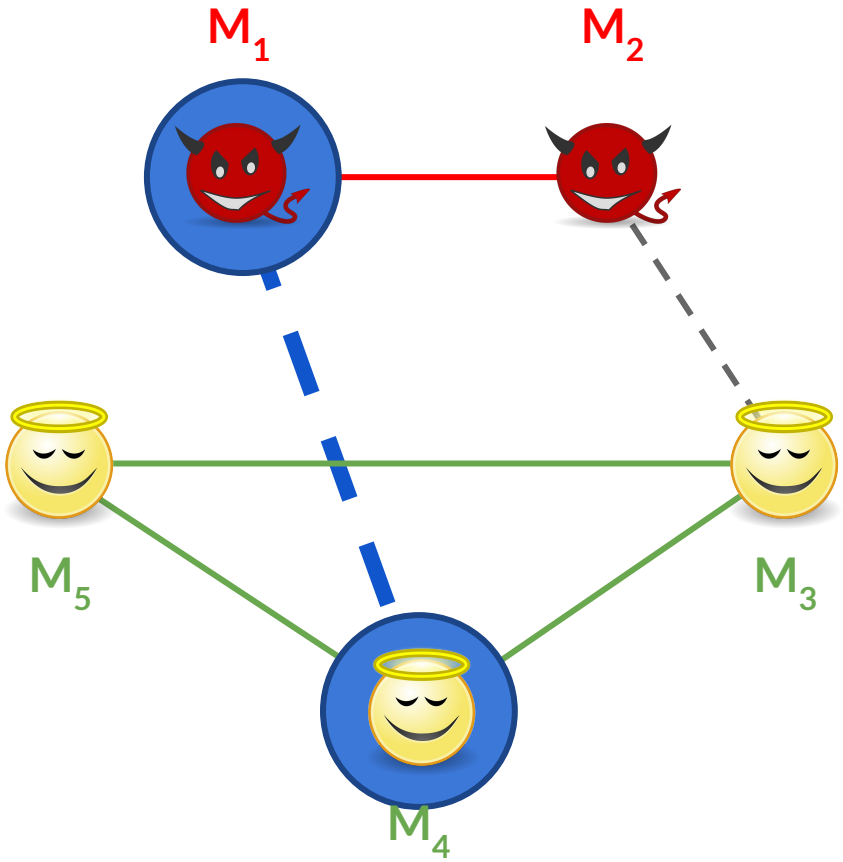
# MIRANDA FROM 10,000 FEET (removing problematic links is a good idea)

Miranda detects problematic pairs involved in active attacks, and removes their shared links, eliminating these attack vectors



# MIRANDA FROM 10,000 FEET (removing problematic links is a good idea)

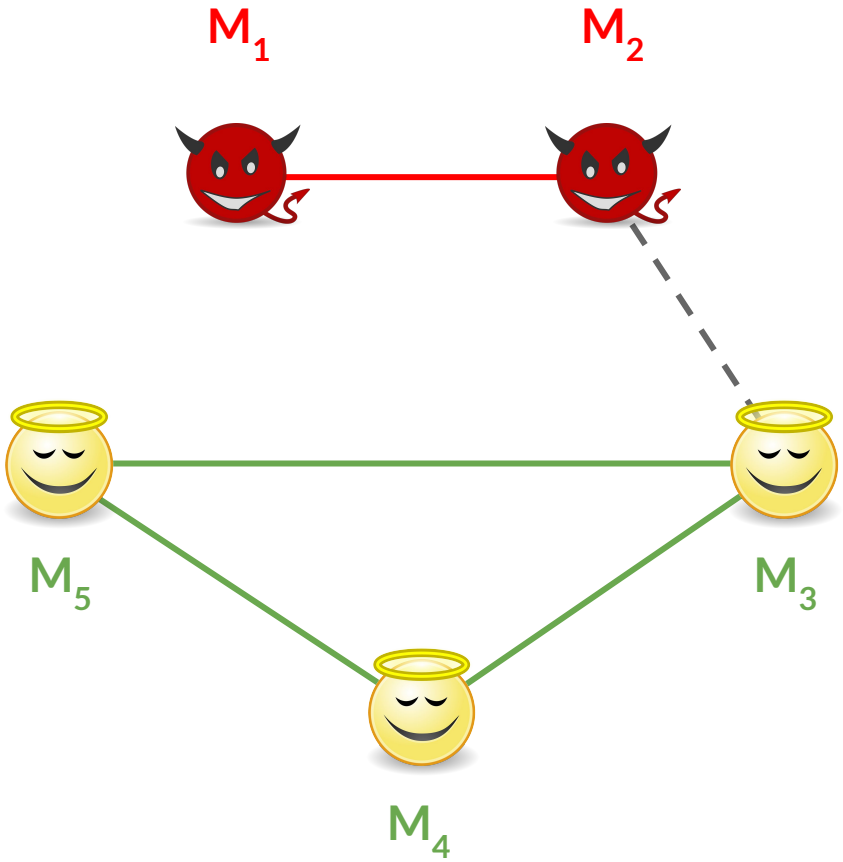
Miranda detects problematic pairs involved in active attacks, and removes their shared links, eliminating these attack vectors



# MIRANDA FROM 10,000 FEET (removing problematic links is a good idea)

The result:

malicious mixes are removed from the system



# MIRANDA'S DESIGN (challenges)

✔ Detect attacks by malicious mixes

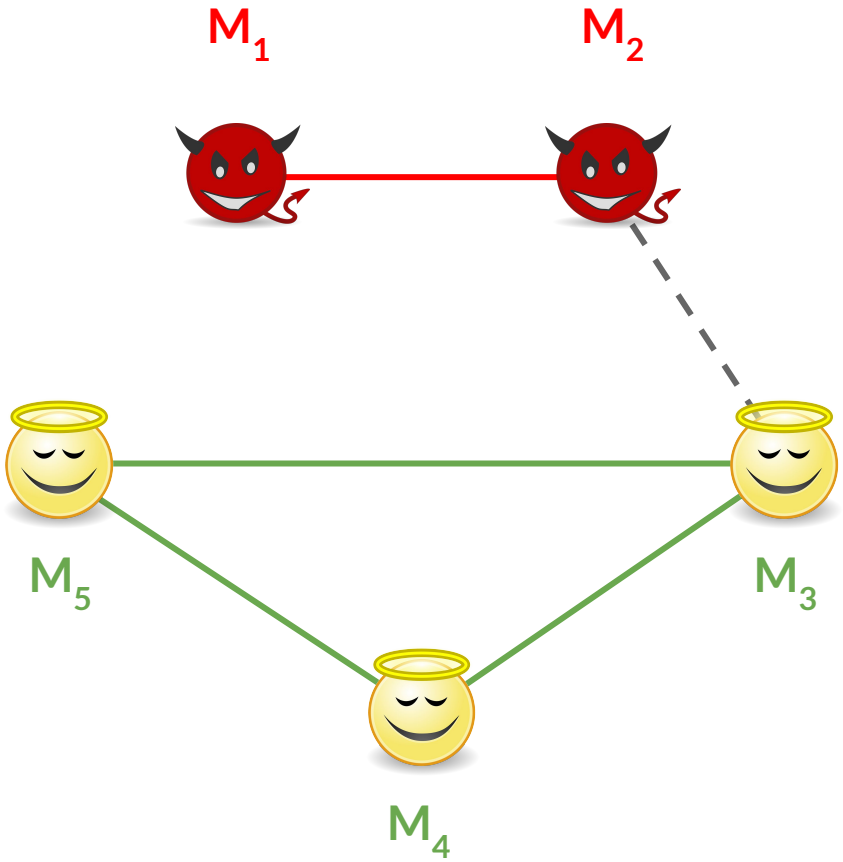
✔ Penalize the malicious mix

➔ **Identify the malicious mix**

# MIRANDA FROM 10,000 FEET (removing problematic links is a good idea)

The result:

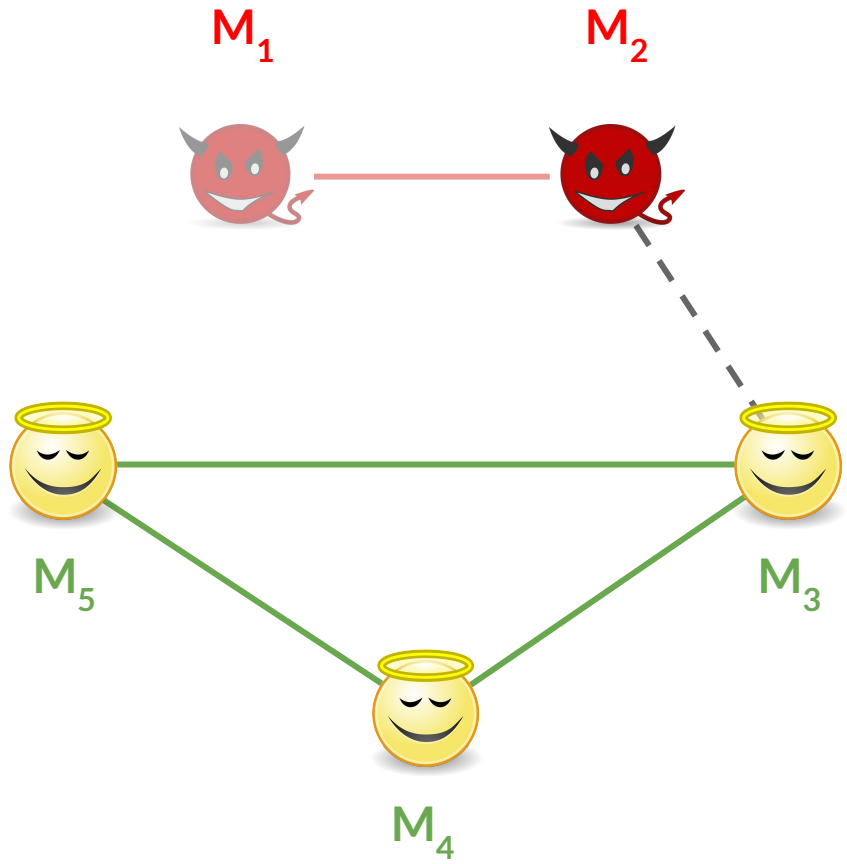
malicious mixes are removed from the system



# MIRANDA FROM 10,000 FEET (removing problematic links is a good idea)

The result:

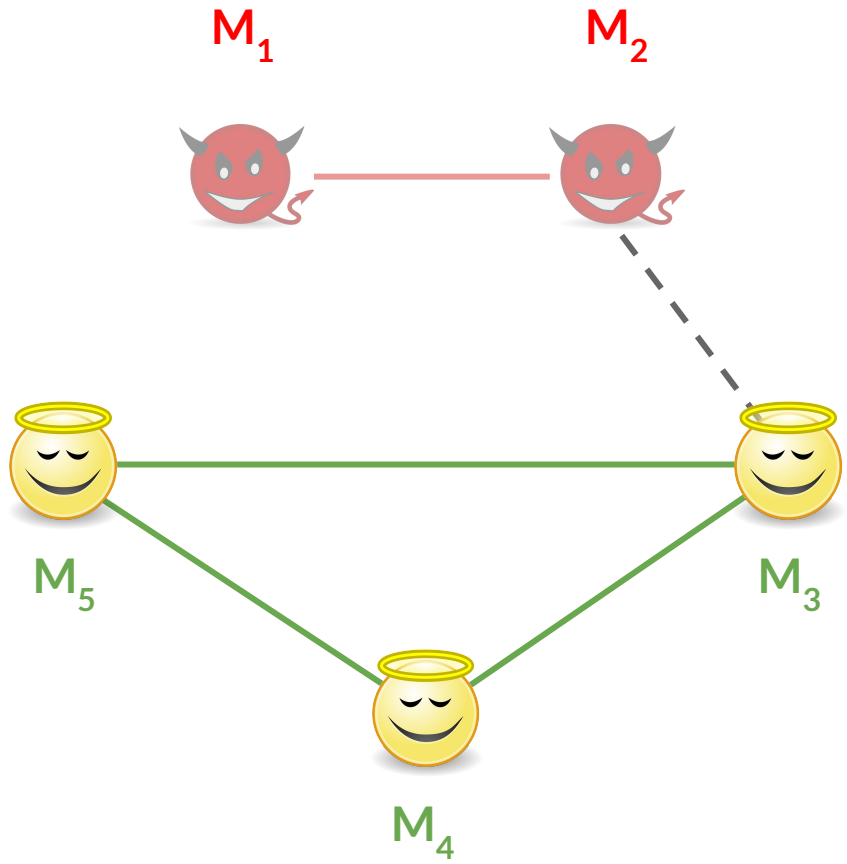
malicious mixes are removed from the system



# MIRANDA FROM 10,000 FEET (removing problematic links is a good idea)

The result:

malicious mixes are removed from the system



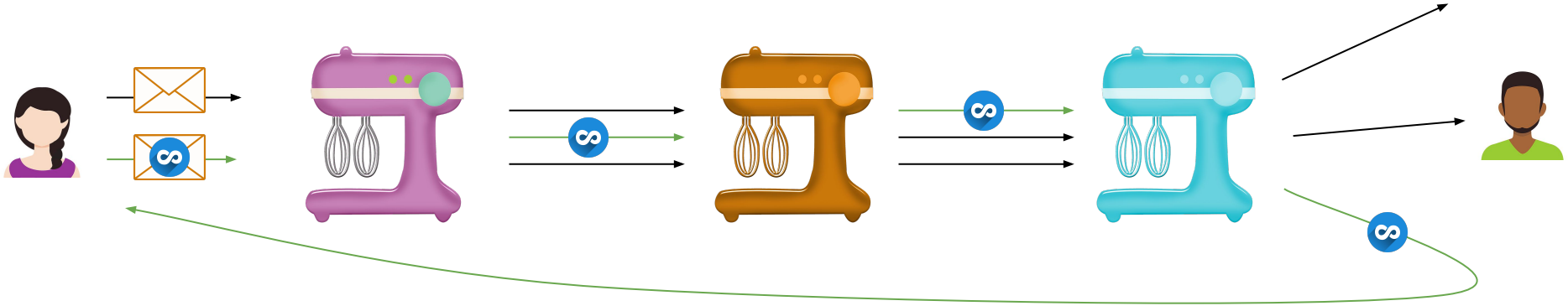
# MIRANDA'S DESIGN (challenges)

➔ **Detect attacks by malicious mixes**

✔ Penalize the malicious mix

✔ Identify the malicious mix

# DETECTING AN ATTACK (using loop messages)



# A LOT MORE IN THE PAPER

- More details
- Community detection techniques: enhanced detection
- Mitigating protocol abuse
- Cascade compilation strategies
- Experimental results

# CONCLUSION

- Miranda is a step in the right direction, but we have not reached the promised land yet
- Future work
  - Complete (provable) security analysis
  - Relax assumptions towards practicality (e.g., churn)
  - Further reduce latency

# THANK YOU

# Questions?

(for example, why the name Miranda?)


**Hemi Leibowitz**

**Leibo.hemi@gmail.com**

**Ania M. Piotrowska**

**a.piotrowska@ucl.ac.uk**

**George Danezis**

**g.danezis@ucl.ac.uk**

**Amir Herzberg**

**amir.herzberg@gmail.com**