

# Minx:

## A Simple and Efficient Mix Packet Format

---

George Danezis  
University of Cambridge, Computer Lab.  
(Thanks to CMI grant)

Ben Laurie  
ALD Ltd.

# Outline

- What is a mix packet format – naïve constructions
- Attacks against mix formats
- Minx – our construction
- Conclusions

## Warning:

This talk abstracts away a lot of boring details – read the paper for full (gory) details.

# What is a mix?

- Building block for anonymous communications.
- Router that hides correspondences.



- Two components:
  - Changes the bit patterns using a secret
  - Disrupts the timing patterns – another subject altogether

# What is a mix packet format?

- Cryptographic format that allows the bit patterns to change – so that it is hard to trace a message.
  - The sender encodes a message.
  - The mix decodes the message and sends it along.
- Naïve construction:



# Anonymous replies

- Similar construction for anonymous reply blocks:



- Support for anonymous replies
  - Indistinguishable from 'normal' messages.
  - Secure against all attacks.

# Multiple Mixes

- Mix networks distribute trust and load.
- Use a chain of mixes instead of one:

A to Mix 1:

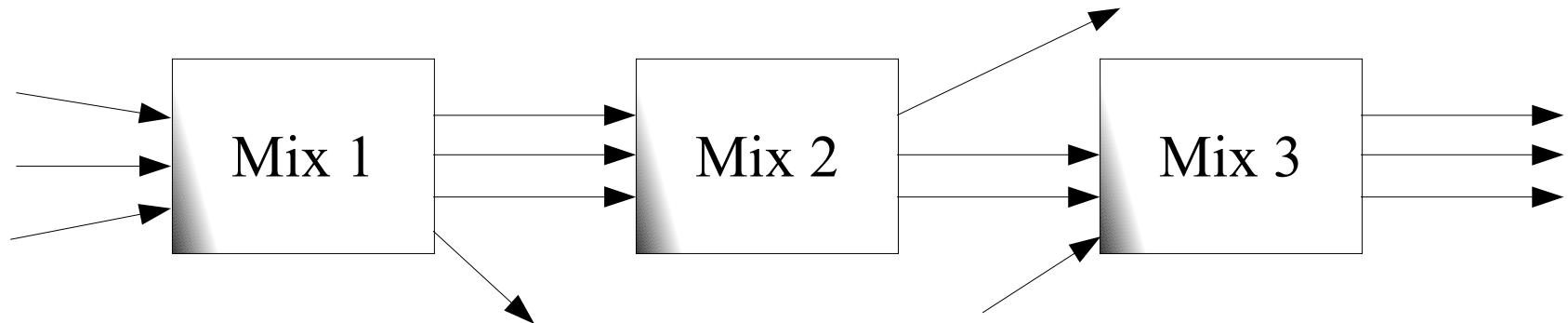
$\{\text{Mix 2}, \{\text{B}, \text{M}\}_{\text{PubK2}}\}_{\text{PubK1}}$

Mix 1 to Mix 2:

$\{\text{B}, \text{M}\}_{\text{PubK2}}$

Mix 2 to B:

M



- Strip the packet layer by layer
- Hide length of path / position

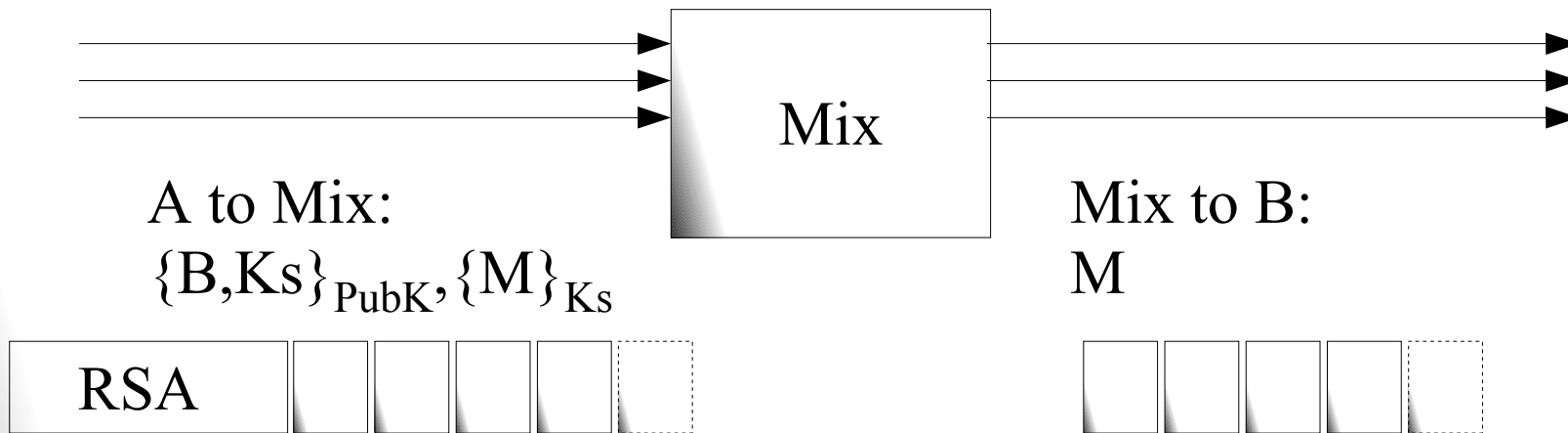
# The real world – red in tooth and claw

---

- Naïve examples so far would not stand a chance.
- Attacks:
  - Attacker can watch all packets coming in and out of mix.
  - The attacker controls a subset of nodes on the path.
  - Attacker can modify/tag messages to trace them – the killer attack.
- Example: Using AES CBC for encryption ...

# What if $\{.\}_K = \text{AES CBC}$

- Use hybrid RSA / AES CBC to encrypt message:



- Attack: flip one bit of some input block



# Surely this is trivial ...

- 
- Other cryptographic constructions:
    - AES Counter – can xor patterns all the way in.
    - RSA/ElGamal – can blind the cipher text
  - As close to adaptive chosen ciphertext attacks as it gets:
    - Construct a number of variants of the message packet and
    - Submit them to the mix / decryption oracle.
    - Additional constraints
      - cannot know the payload of replies – therefore cannot use integrity protection,
      - mix could be the bad guy!

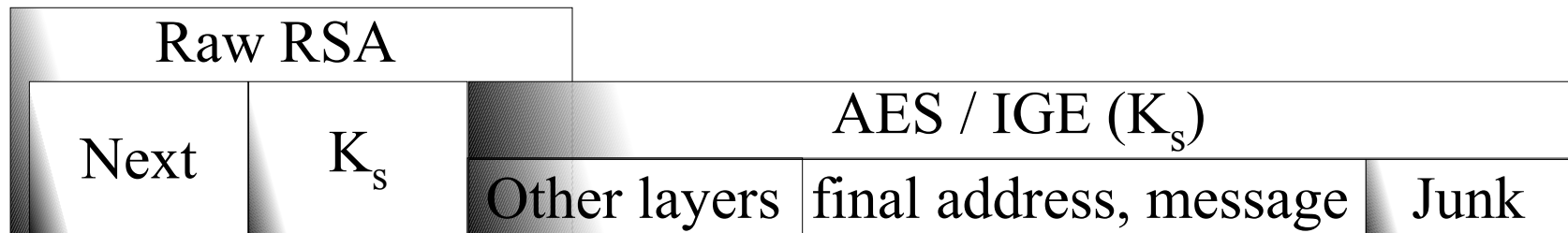
# Minx: design principles

---

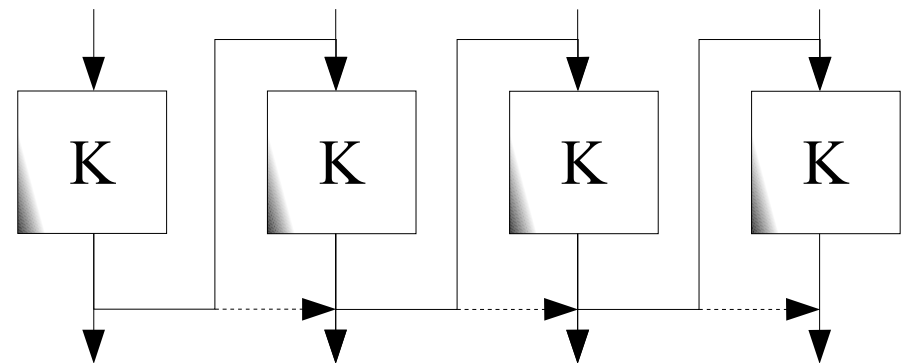
- Do not use any integrity primitives – they leak information to intermediary mixes.
- All messages in the network are indistinguishable from random.
- Fragile messages to foil tagging attacks:
  - When modified all useful information is destroyed!  
(Useful = Final address, message payload)
  - Keep track of messages routed for duplicate prevention.

# The anatomy of Minx

- One layer of Minx:

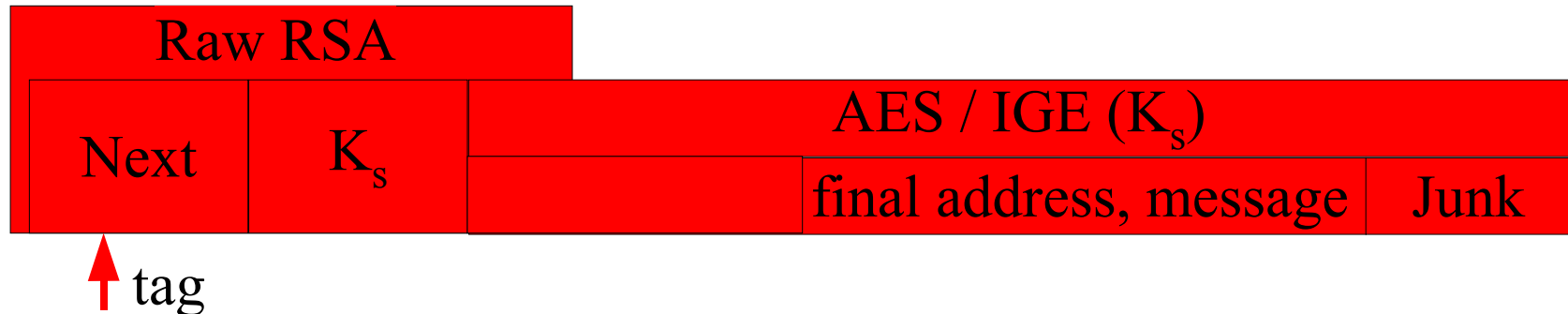


- Standard building blocks: raw RSA / AES
- Non standard mode: infinite garble extension (IGE)
  - Like CBC but...
  - Perfect error propagation



# Tagging the header

- One layer of Minx:



- $K_s$  cannot be recovered – wrong decryption.
- Packet is not discarded but routed to a random address.
- Correct packer is not recorded as processed.

# Tagging the body (other headers)

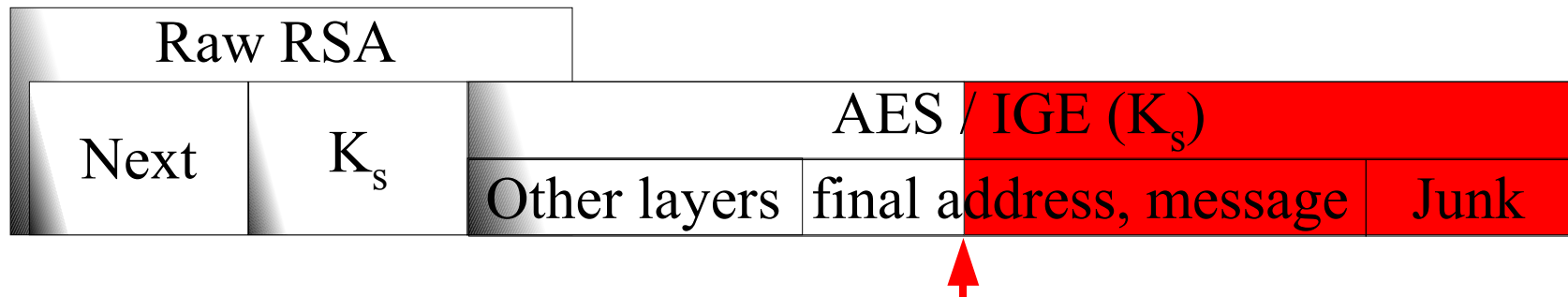
- One layer of Minx:



- The rest of the message is corrupt – will be routed!
- Final address / payload destroyed forever.
- Message register – no message with key K<sub>s</sub> will be decrypted again.

# Tagging the body (payload)

- One layer of Minx:



- Oops – half the payload is random (attack!)
- Easy to fix: use AONT for all payload
- If any is modified all is modified – no leak.
  - IGE makes sure that at least a block will be modified.

# Conclusions

---

- Read the paper – key privacy, RSA decryption must always be random, routing information must be random, avoid ghost messages. So many things can go wrong!
- Yes it also works for replies – with no modification.
- Fragile messages self-destruct instead of compromising anonymity.
- Covert traffic is generated when attack is going on.
- Chosen ciphertext security in practice/other problems.
- We can do it – but where are the proofs of security?