An Address "Ownership" Problem in IPv6

How to handle authorization in IPv6 signalling mechanisms that affect routing

Pekka Nikander
Ericsson Research NomadicLab
Pekka.Nikander@nomadiclab.com

`draft-nikander-ipng-address-ownership-00.txt`
`draft-nikander-ipng-pbk-addresses-00.txt (to appear)`

Overview

- Problem statement
- Current extent of the problem
- Attack example: "Future" stealing
- Hardest case: Mobile Networks
- Ingredients for a partial solution
  - Combining OTP + host ID as a crypto token
  - Relying on routing structure
  - Wrapping up the solution
- Summary
- What next?

Problem statement

- Who is *authorized* to change routing information for a specified IP address or address prefix?
  - Focus: temporary changes e.g. for mobility
  - Scope: any address/host in the Internet
- Answer: whoever "owns" or "controls" the address
  - * (Yes, this is a tautology, but restating a problem often helps)
- Restated problem:
  How do you *show* that you "own" an IP address?
  - More specifically: that you "own" it now and in the (near) *future* as well
- NOTE! Authentication (as per IPsec) is not sufficiently alone; having an IPsec association with a host is *not* a proof that the host is fully honest and competent

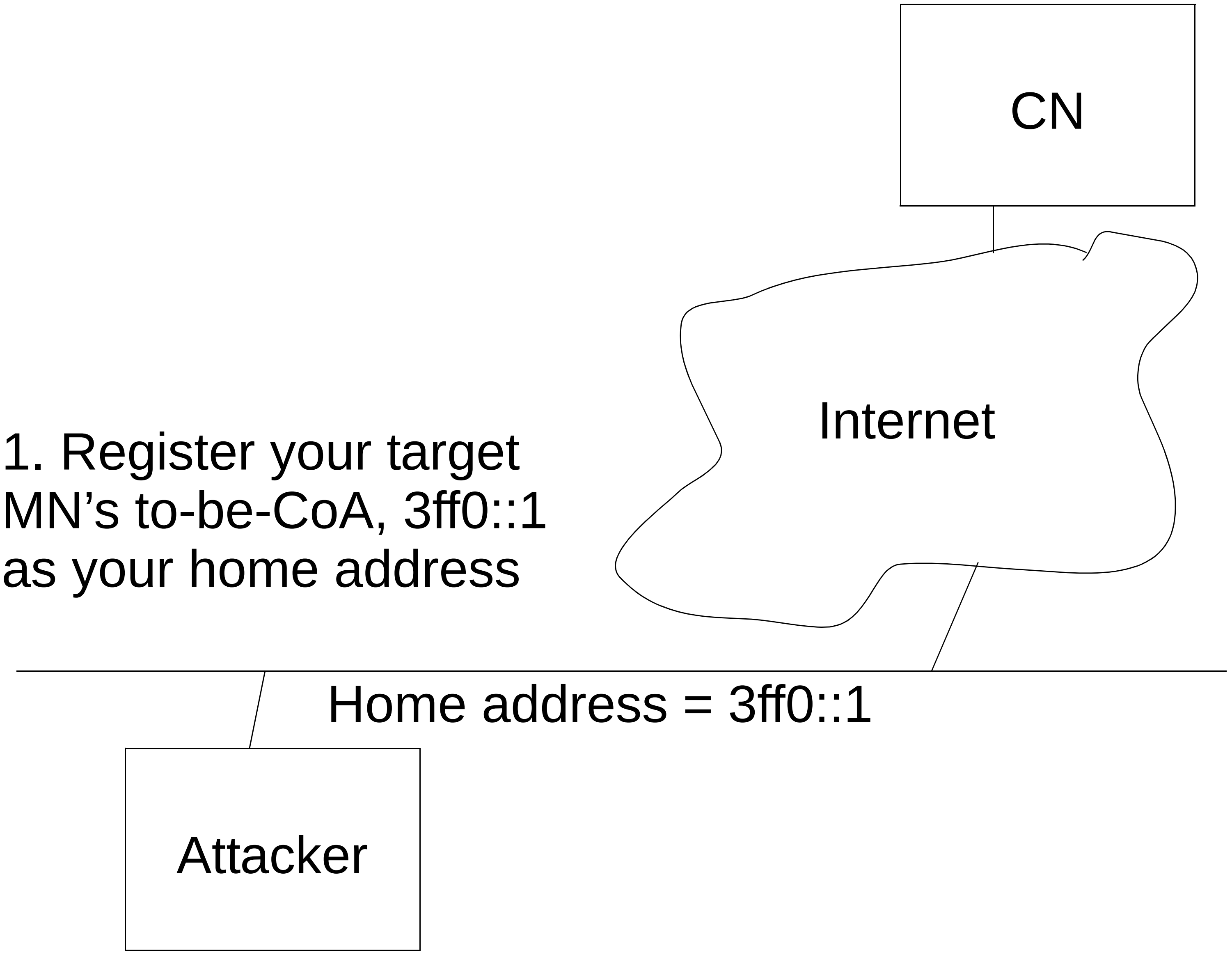
Current extent of the problem

Affected Entered by	Any size prefix / router / host	Any size prefix / host only	Single address only	Single reply packet only
Local administrator	Basic routing info Generic tunnels			
Any host on local link		Router discovery	ICMP Redirect	
Any trusted host in the Internet	Router renumbering	IPsec tunnels		
Any host in the Internet (that you accept IPsec from)			Mobile IPv6 Binding Updates	Routing Header

- Possible new issues in near future:
  SCTP, Inverse ND, SeaMoby context transfer?

Attack example: "Future" stealing

- Redirect traffic sent to an address that you anticipate that your target will be using in the future
- A hypothetical example: divert Mobile IPv6 by creating a Binding for a CoA that your target is likely to use



CN

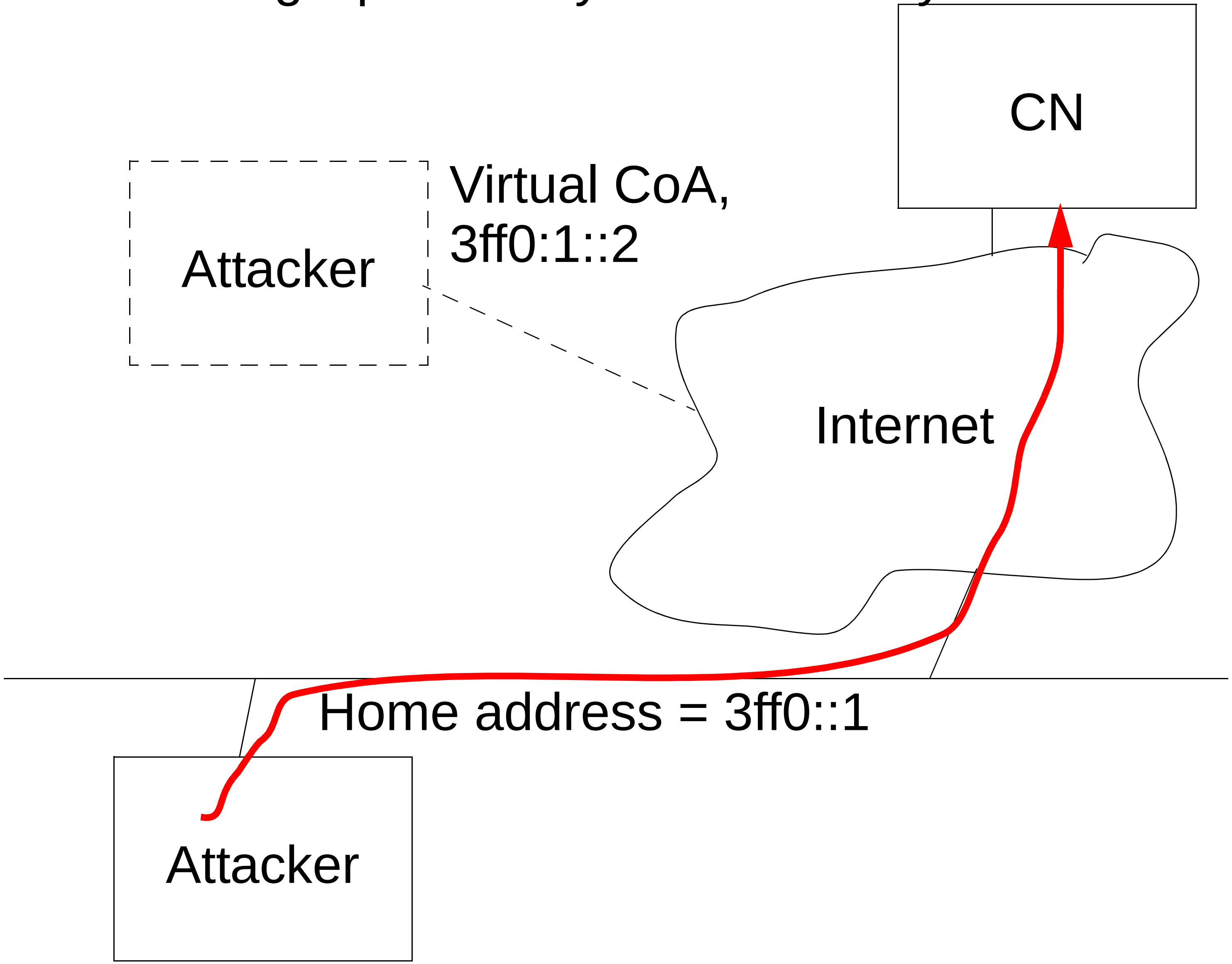
Internet

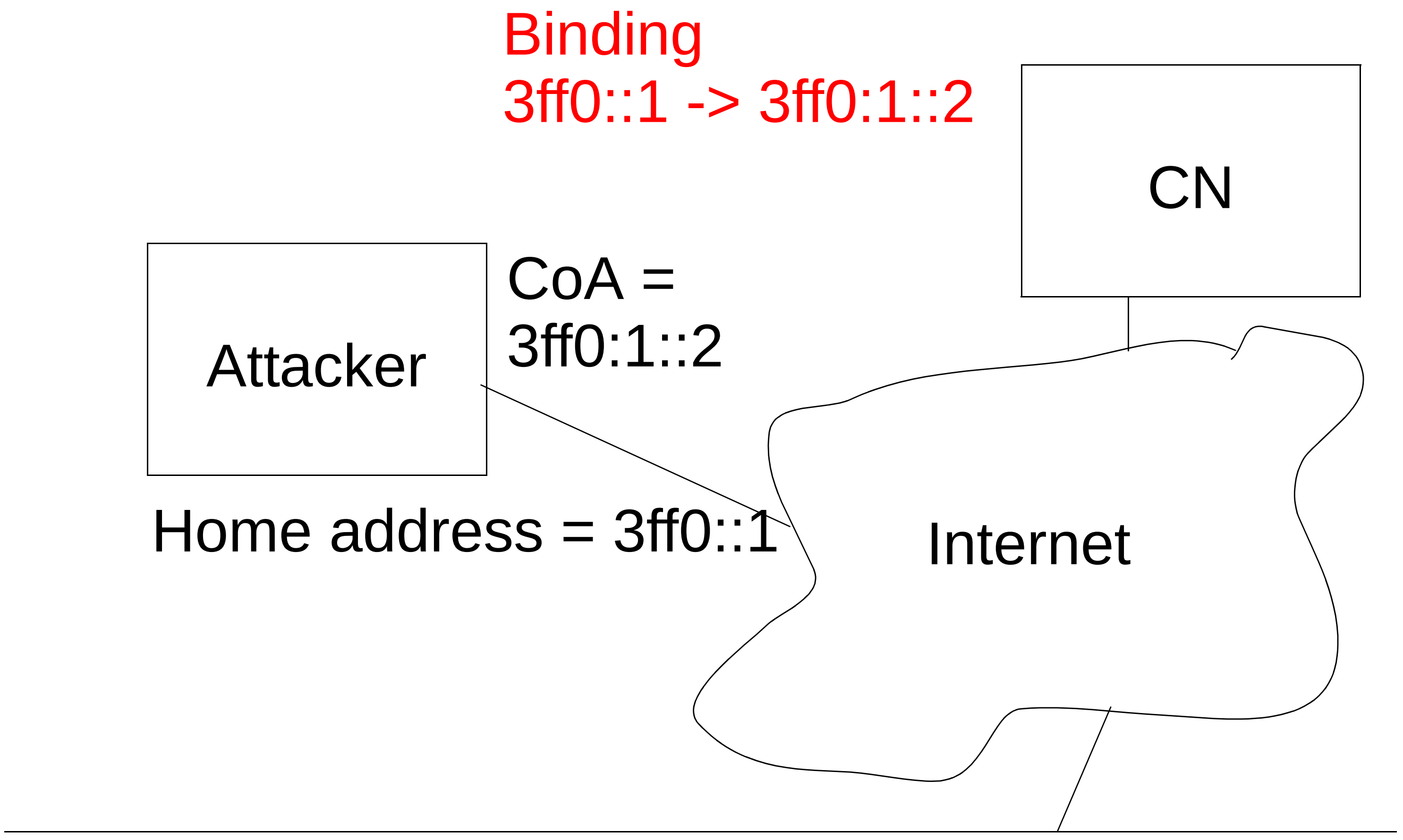
1. Register your target
MN’s to-be-CoA, 3ff0::1
as your home address

Home address = 3ff0::1

Attacker

2. Send Binding Update as you move away





Binding
3ff0::1 -> 3ff0:1::2

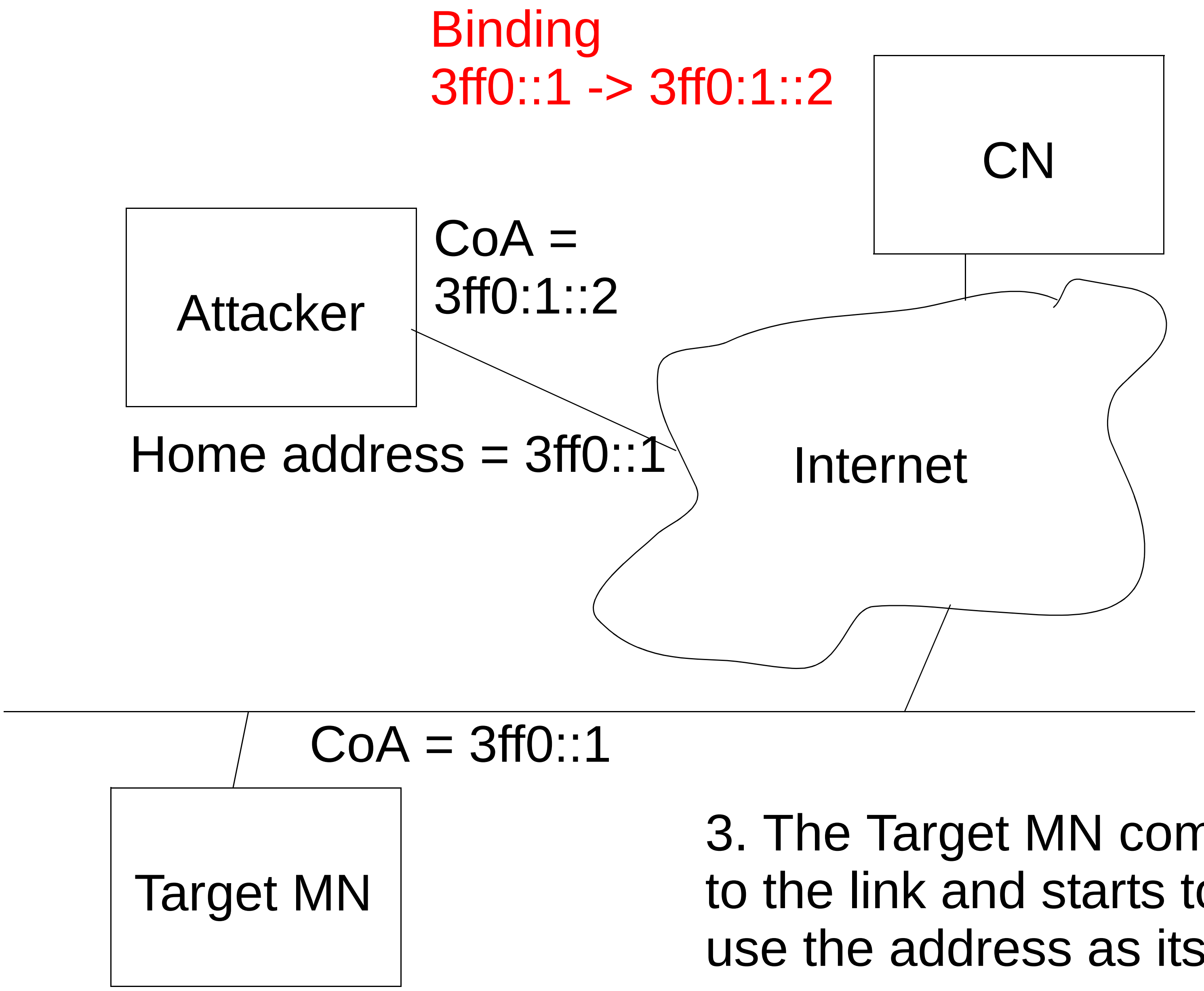
CN

Attacker

CoA =
3ff0:1::2

Home address = 3ff0::1

Internet



Binding
3ff0::1 -> 3ff0:1::2

CN

Attacker

CoA =
3ff0:1::2

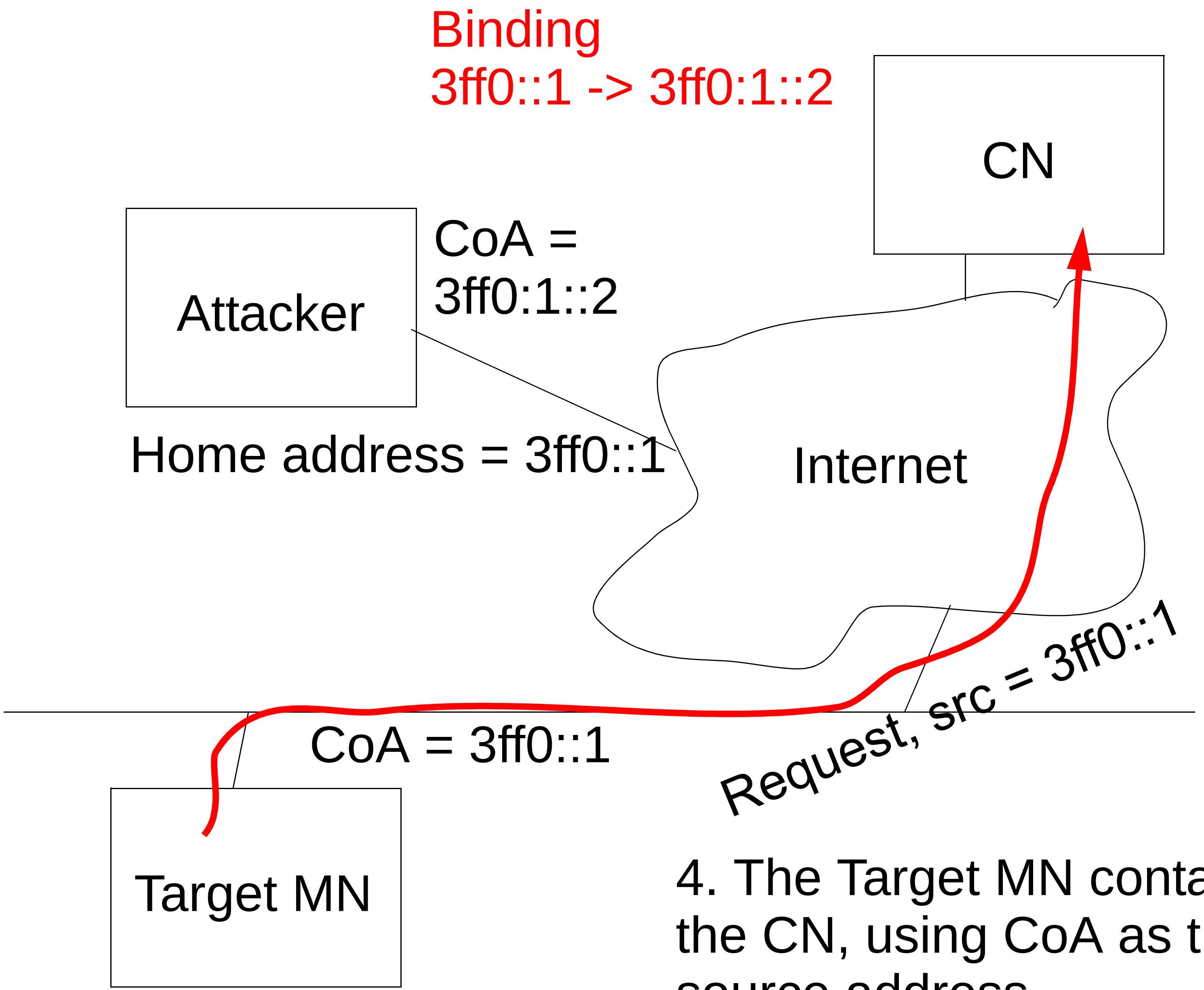
Home address = 3ff0::1

Internet

CoA = 3ff0::1

Target MN

3. The Target MN comes
to the link and starts to
use the address as its CoA



Binding
3ff0::1 -> 3ff0:1::2

CN

Attacker

CoA =
3ff0:1::2

Home address = 3ff0::1

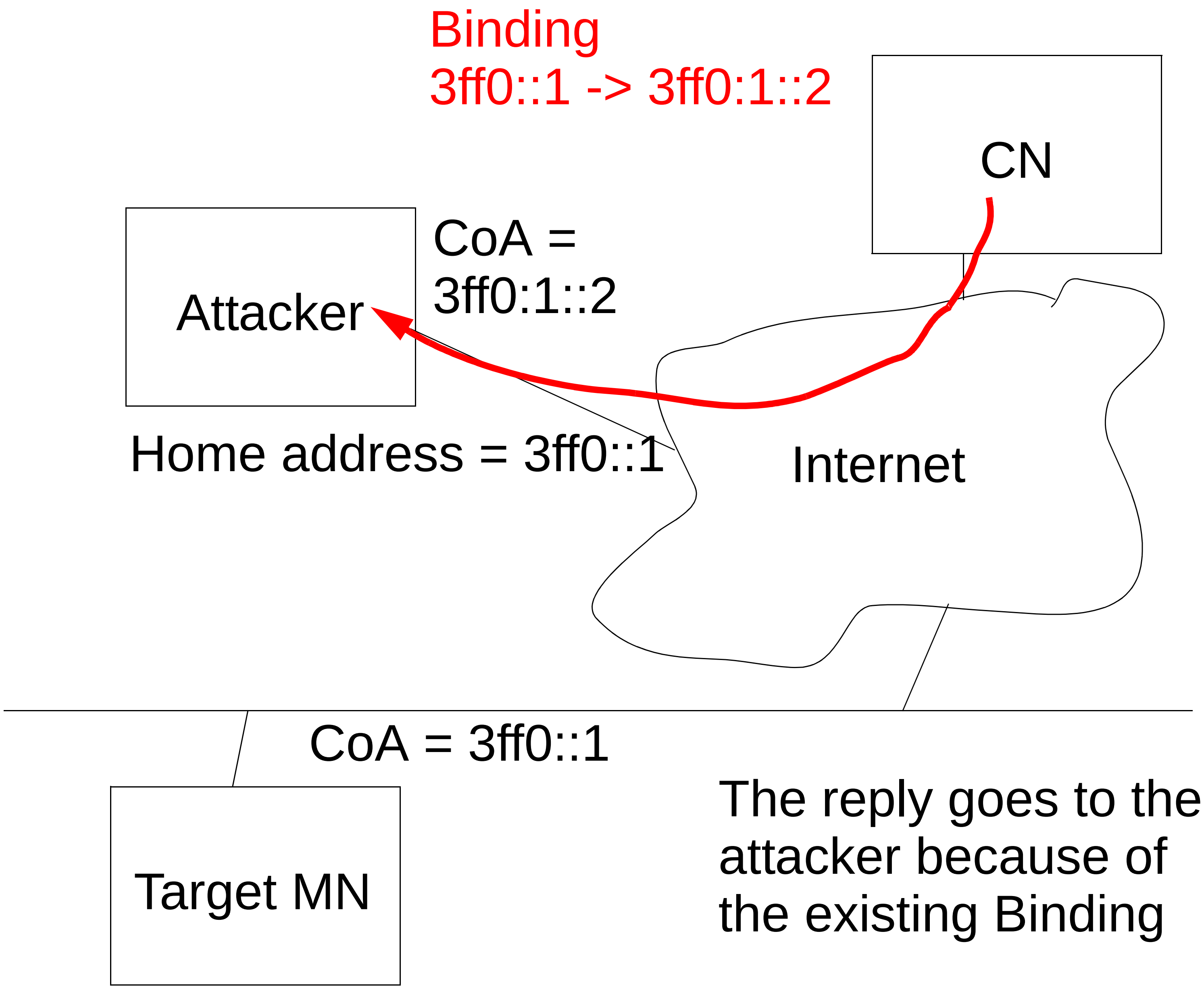
Internet

CoA = 3ff0::1

Request, src = 3ff0::1

Target MN

4. The Target MN contacts the CN, using CoA as the source address



Binding
3ff0::1 -> 3ff0:1::2

CN

CoA =
3ff0:1::2

Attacker

Home address = 3ff0::1

Internet

CoA = 3ff0::1

Target MN

The reply goes to the attacker because of the existing Binding

Hardest case: Mobile Networks

- Address ownership for single addresses may be workable (a proposed solution to follow)
  - You can challenge the "owner" of the address to show that it really controls the address right now
- Address ownership for mobile subnets seems much harder
  - Problem 1: How do you challenge the router to show that it owns all of the subnet it claims to own?
  - Problem 2: What are the security implications to the hosts that move along with the mobile subnet?

Ingredients for a partial solution

- Check that you can reach the "owner"
  - Send a challenge to the address
  - Believe only if you get a corresponding reply
- Use random addresses against future address stealing
  - If the attacker cannot anticipate your address, it has much harder time to establish a binding before you
- Protect the random addresses using an OTP like mech.
  - Generate the random part of the address through a series of hashes, and reveal them in reverse order
- In the process, optionally bind a temporary (PBK) public key to the address, using the address as a crypto token
- The following description is *simplified*, the actual protocol is presented in the draft-to-come

Combining OTP + host ID as a crypto token

- First level construction

  host ID = HASH(public key || random)

- By revealing random, the user of the host ID shows
  - that it generated the host ID since it knows random
  - that it intends to use the public key
- Problem: this works only once, you have to use expensive public key crypto after revealing
- Second level construction

  H_N = HASH(public key || random number)

  H_i = HASH(public key || H_{i+i})

  host ID = H_0 = HASH(public key || H_1)

- Now you can show that you generated H_0, ..., H_N one by time without using public key crypto

Relying on routing structure

- Two parties: a *claimant* wanting to show that the it "owns" an address, and a *verifier* verifying the claim

1. Claimant sends the public key and H_1 to the verifier
2. Verifier verifies that host ID = HASH(public key || H_1), and if so, creates a challenge

   C = HASH(nonce || H_1),

   and sends it back to the verifier
3. Claimant gets challenge and creates response

   R = HASH(C || H_1),

   optionally signed with its public key
4. Verifier verifies the response and optionally checks the signature using the claimant's public key

- Challenge/response checks reachability, host ID provides public key allowing optional signature check

Wrapping up the solution

- Optional public keys as in PBK / HIP
- Random host IDs to protect against the "future" attack
- Public key bound to host ID through a hash
  - The MAC address can also be bound to the host ID in the same way, if that provides better protection
- Series of hashes to repeatedly show local "ownership"
- Challenge/response used to check current reachability

  `http://www.tml.hut.fi/~pnr/publications/`
  `draft-nikander-ipng-pbk-addresses-00.txt`
- Need to consider how to apply this to Mobile IPv6
  - Need to find out the real security requirements
- Mandatory claim: Ericsson has filed a patent application which may be relevant to some of the issues presented

Summary

- Address "ownership" is a real problem already present in several signalling functions within IPv6
- The question is about *authorization*: who is *entitled* to change routing information wrt. a specific address
  - Authorization is always application specific; here the aplication is IPv6 signalling affecting routing
- We are working on a solution that
  - Creates a binding from an address to a public key
  - Uses routing infrastructure for reachability check
  - Uses an OPIE like series of hash values to block DoS in IPv6 Duplicate Address Detection (DAD)
  - Uses random addresses to block the "future" attack
- We are looking at how to apply this to Mobile IPv6

What next?

- A solution for Mobile IPv6 specifically
  - A proposal within the next couple of weeks
- Further clarification of the scope of the problem
  - More work needed at least for SCTP and Inverse ND, possibly other issues
  - **draft-nikander-ipng-address-ownership** into a Informational RFC?
  - Volunteers?
- Work for a generic solution for address ownership?
  - Is the 63-bit binding between host ID and a public key of any real use?
  - How about closing the DoS attack in DAD?