SECURING IPV6

Cynthia Omauzo

DREU Summer 2015

ARP in IPv4

What is ARP

Address Resolution Protocol. This protocol is used to map an IP address to a physical machine address (or MAC address)

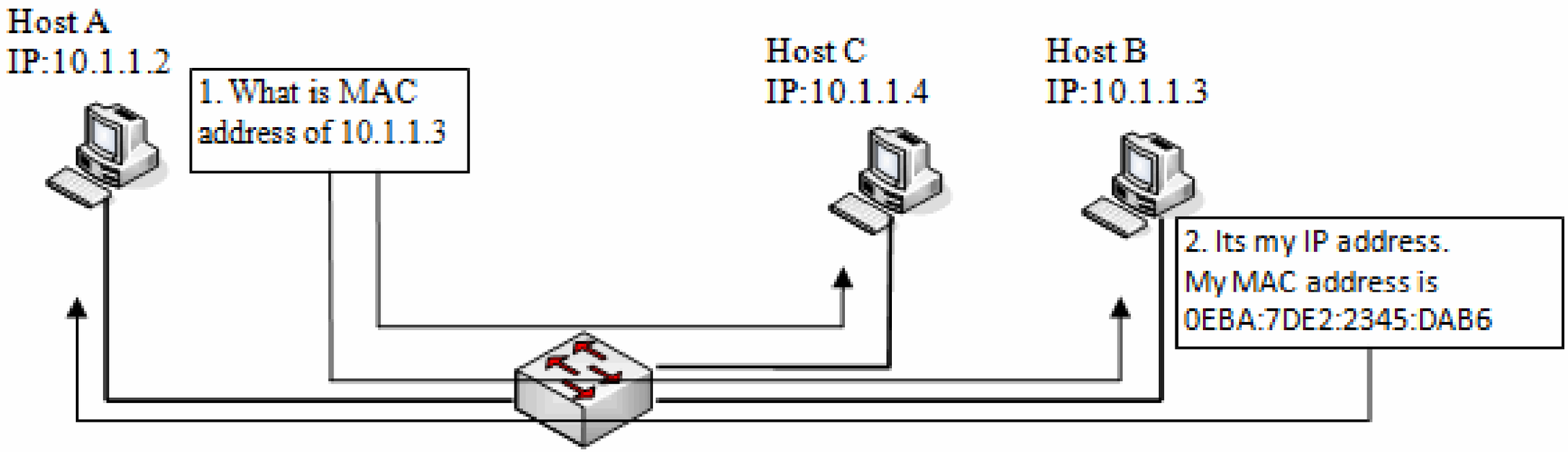
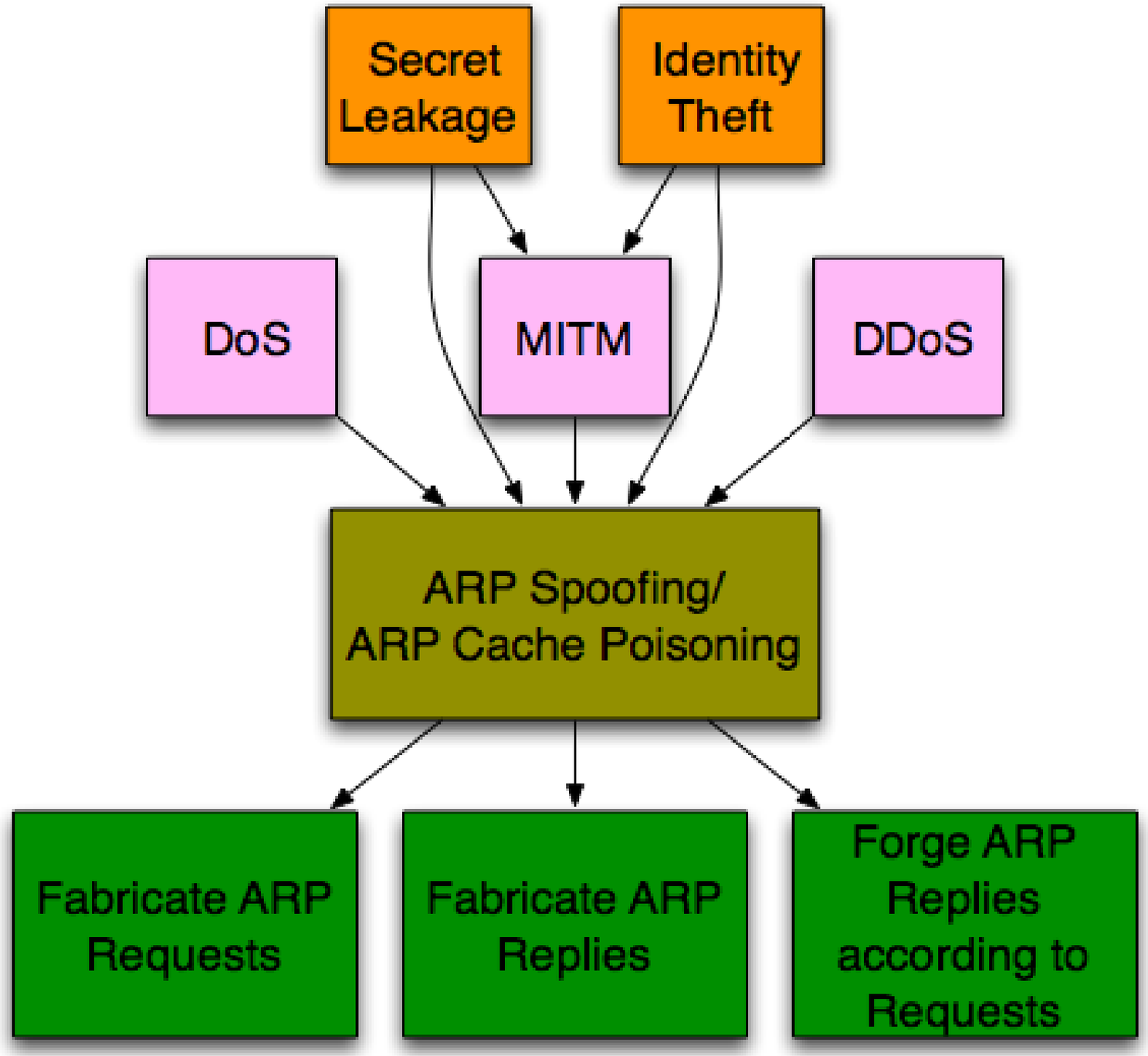
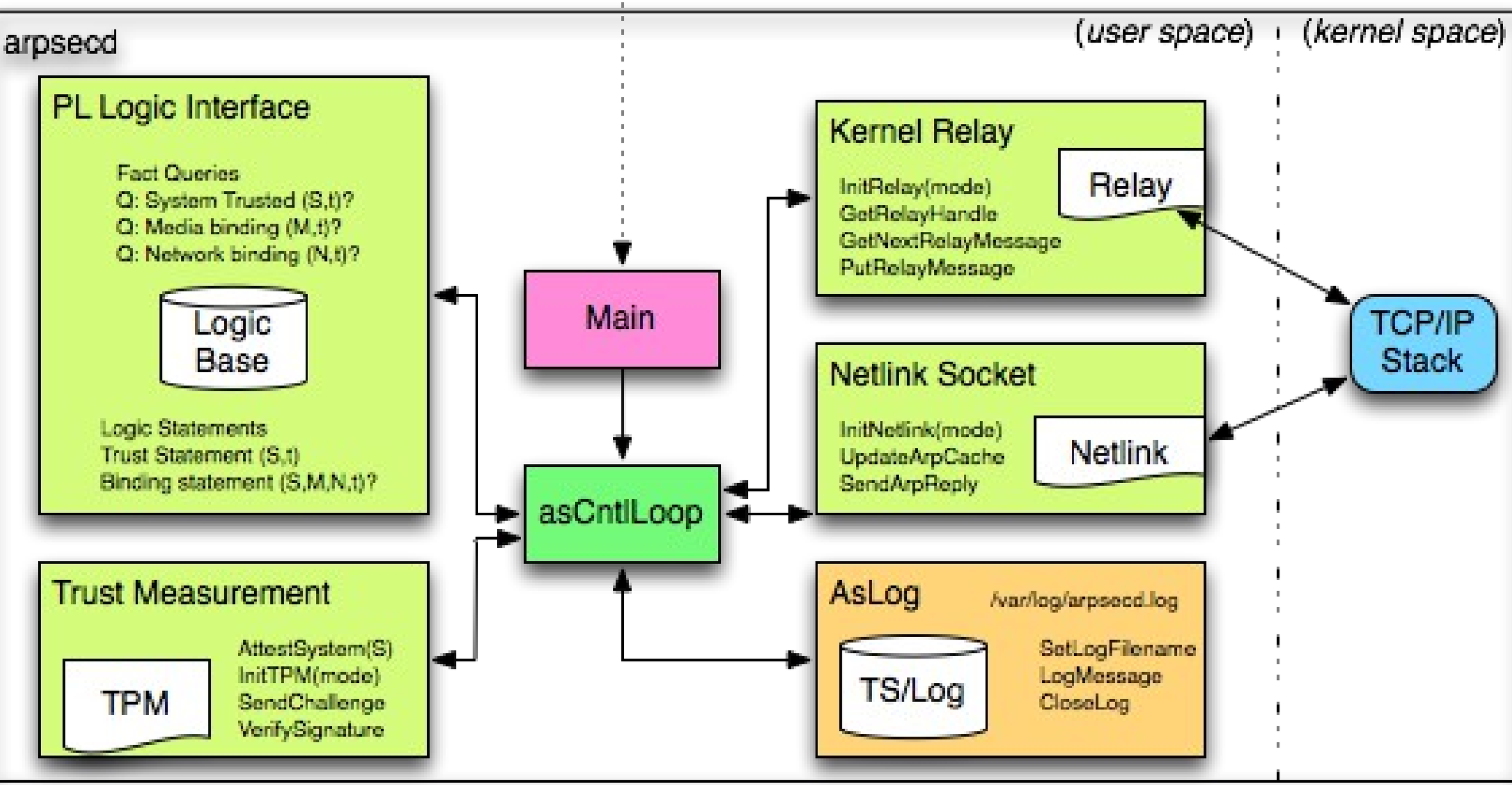


Figure 2.5. ARP operation on the LAN

ARP Vulnerabilities



ARPsec


arpsecd
(user space)
(kernel space)
PL Logic Interface
Fact Queries
Q: System Trusted (S,t)?
Q: Media binding (M,t)?
Q: Network binding (N,t)?
Logic Base
Logic Statements
Trust Statement (S,t)
Binding statement (S,M,N,t)?
Main
asCntlLoop
Kernel Relay
InitRelay(mode)
GetRelayHandle
GetNextRelayMessage
PutRelayMessage
Relay
Netlink Socket
InitNetlink(mode)
UpdateArpCache
SendArpReply
Netlink
TCP/IP Stack
Trust Measurement
TPM
AttestSystem(S)
InitTPM(mode)
SendChallenge
VerifySignature
AsLog
/var/log/arpsecd.log
TS/Log
SetLogFilename
LogMessage
CloseLog

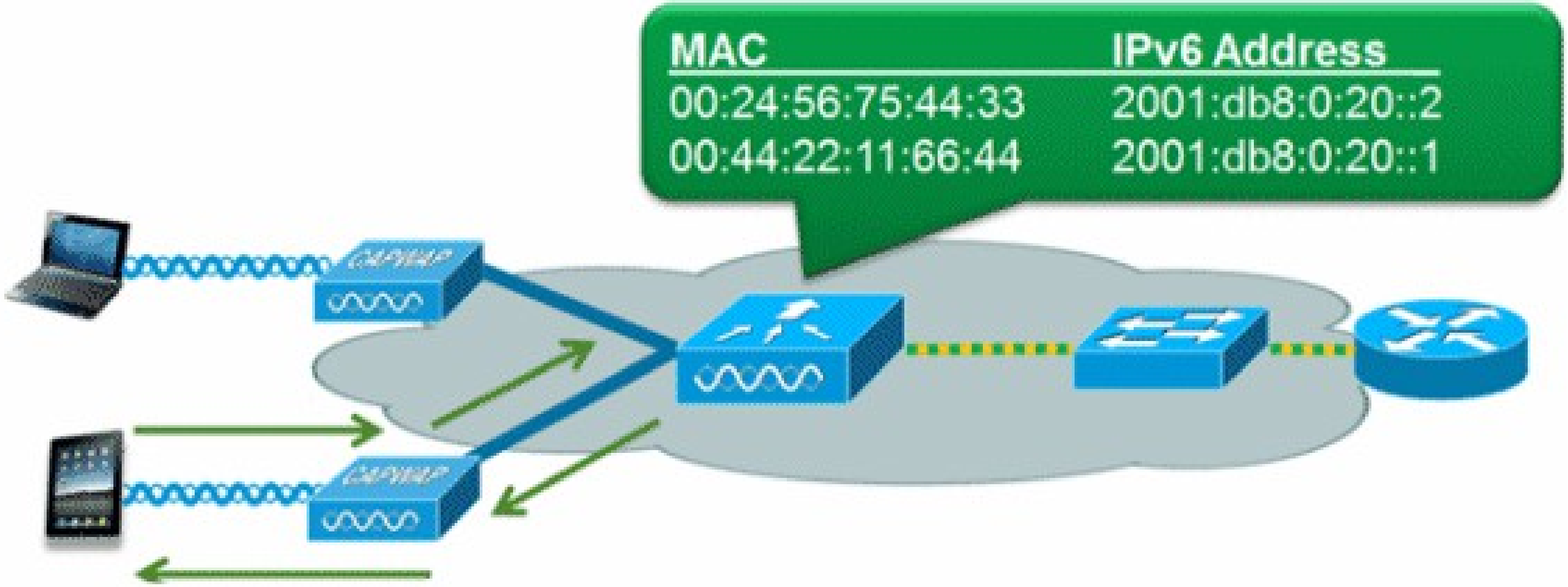
More About TPM

Trusted Platform Model is a cryptographic chip embedded in motherboards.

NDP in IPv6

What is Neighbor Discovery

- The function of Neighbor Discovery (ND) is for a host to learn the IPv6 addresses of its neighbors.



NDP in IPv6

- Neighbor Discovery Protocol maintains the same basic principles of ARP in IPv4, but has some important modifications.

- ND is a messaging protocol. It is a group of activities that are performed through the exchange of messages.

How NDP Is Used

Used by nodes to:

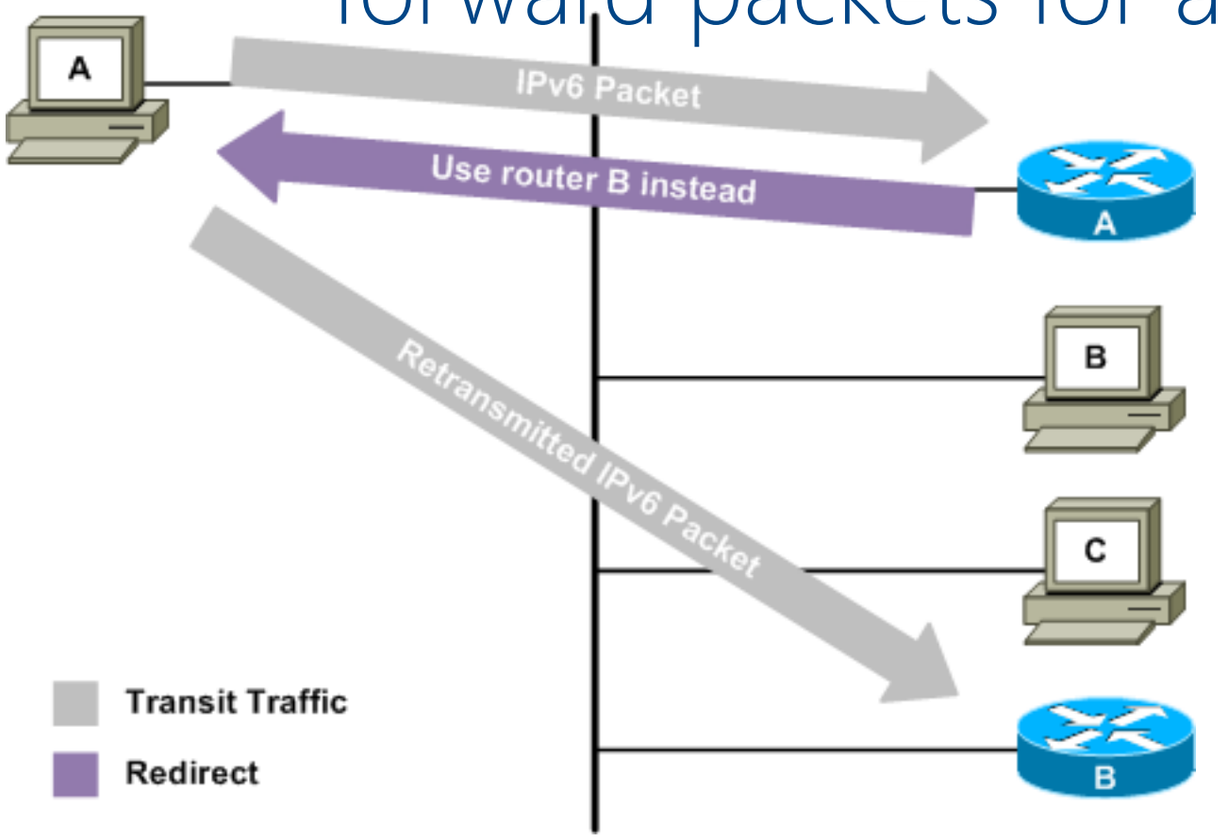
- resolve the link-layer address of a neighboring node to which an IPv6 packet is being forwarded.
- determine when the link-layer address of a neighboring node has been changed
- determine whether a neighbor is still reachable

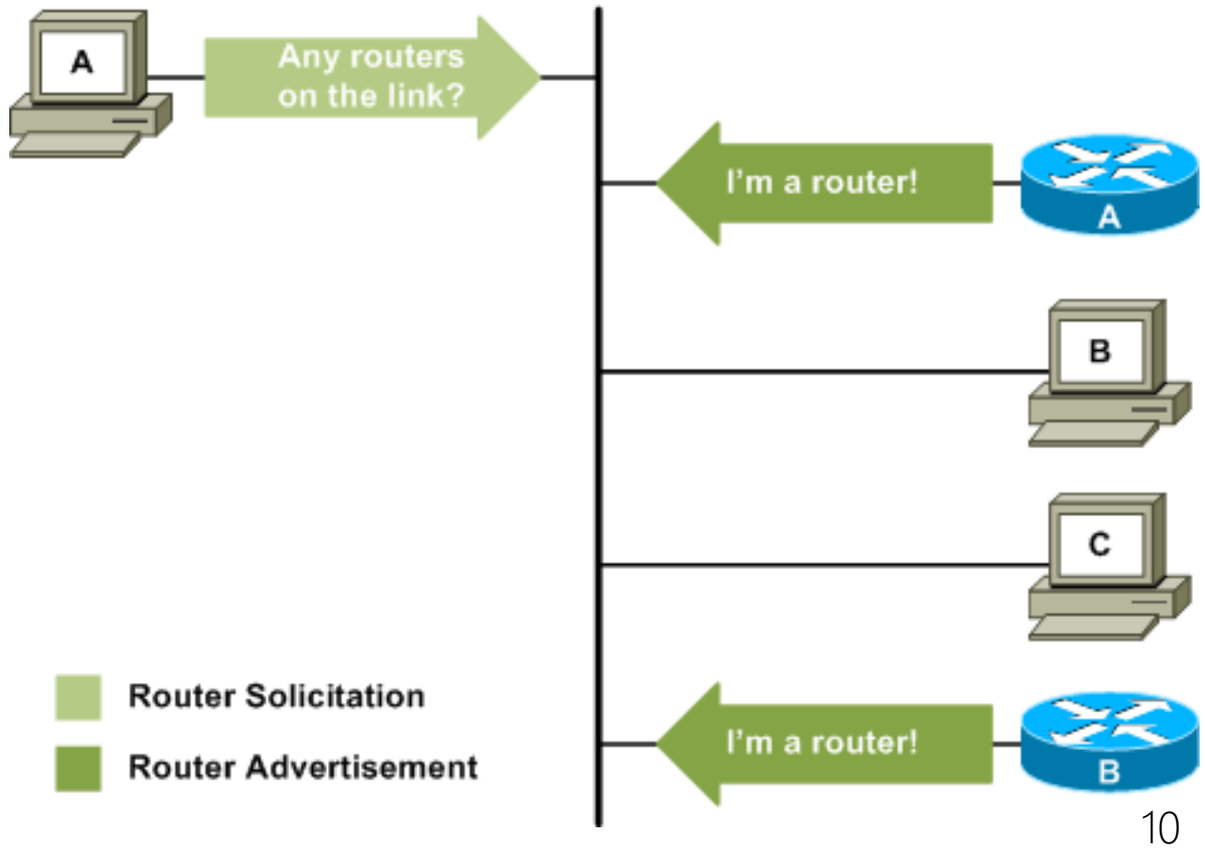
Used by hosts to:

- discover neighboring routes
- auto configure addresses, address prefixes, routes, and other configuration parameters

Used by routers to:

- advertise their presence, host configuration parameters, routes, and on-link prefixes
- inform hosts of a better next-hop address to forward packets for a specific destination



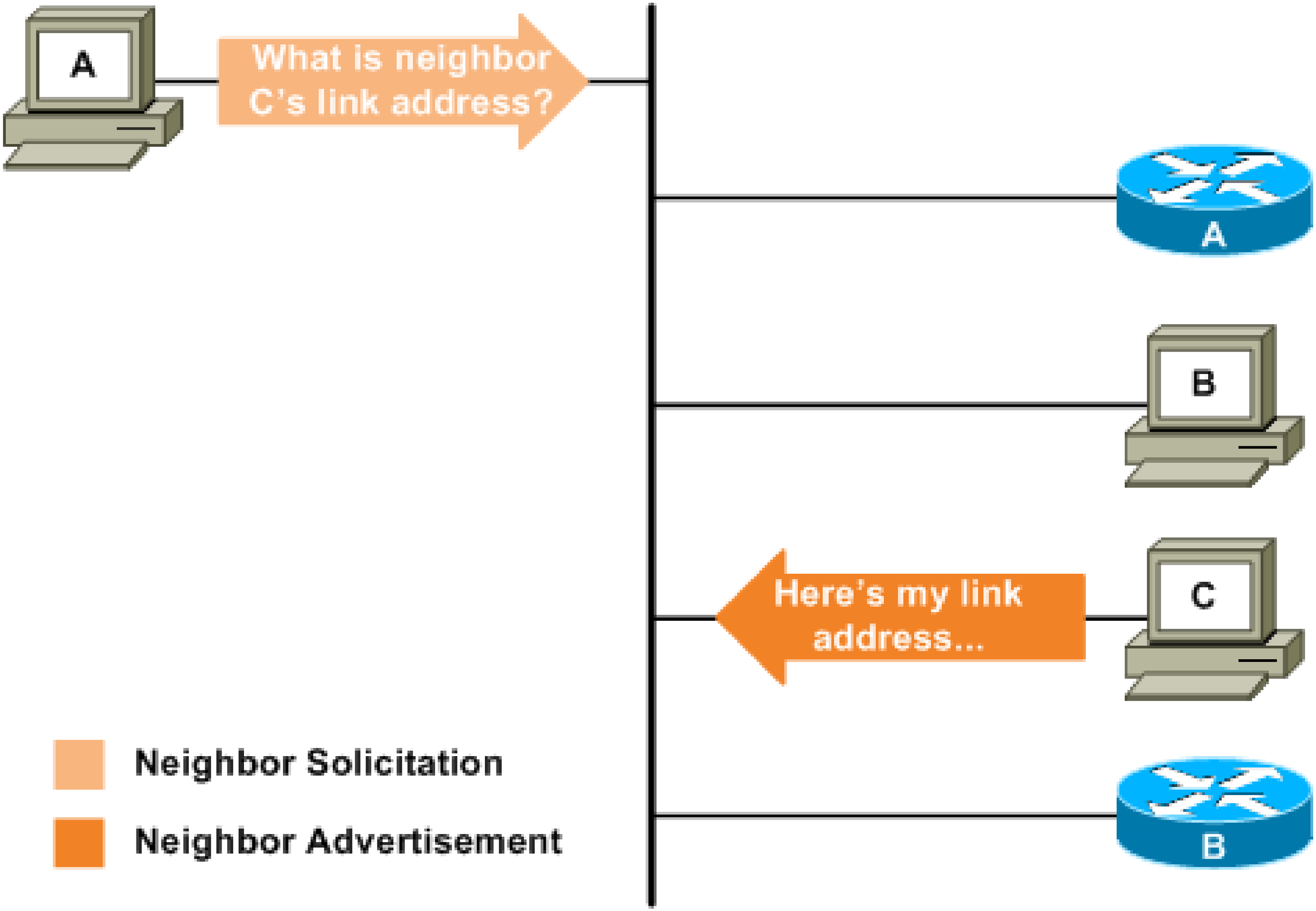


NDP Messages

There are five different ND messages:

- Router Solicitation
- Router Advertisement
- **Neighbor Solicitation**
- **Neighbor Advertisement**
- Redirect

NDP Messages

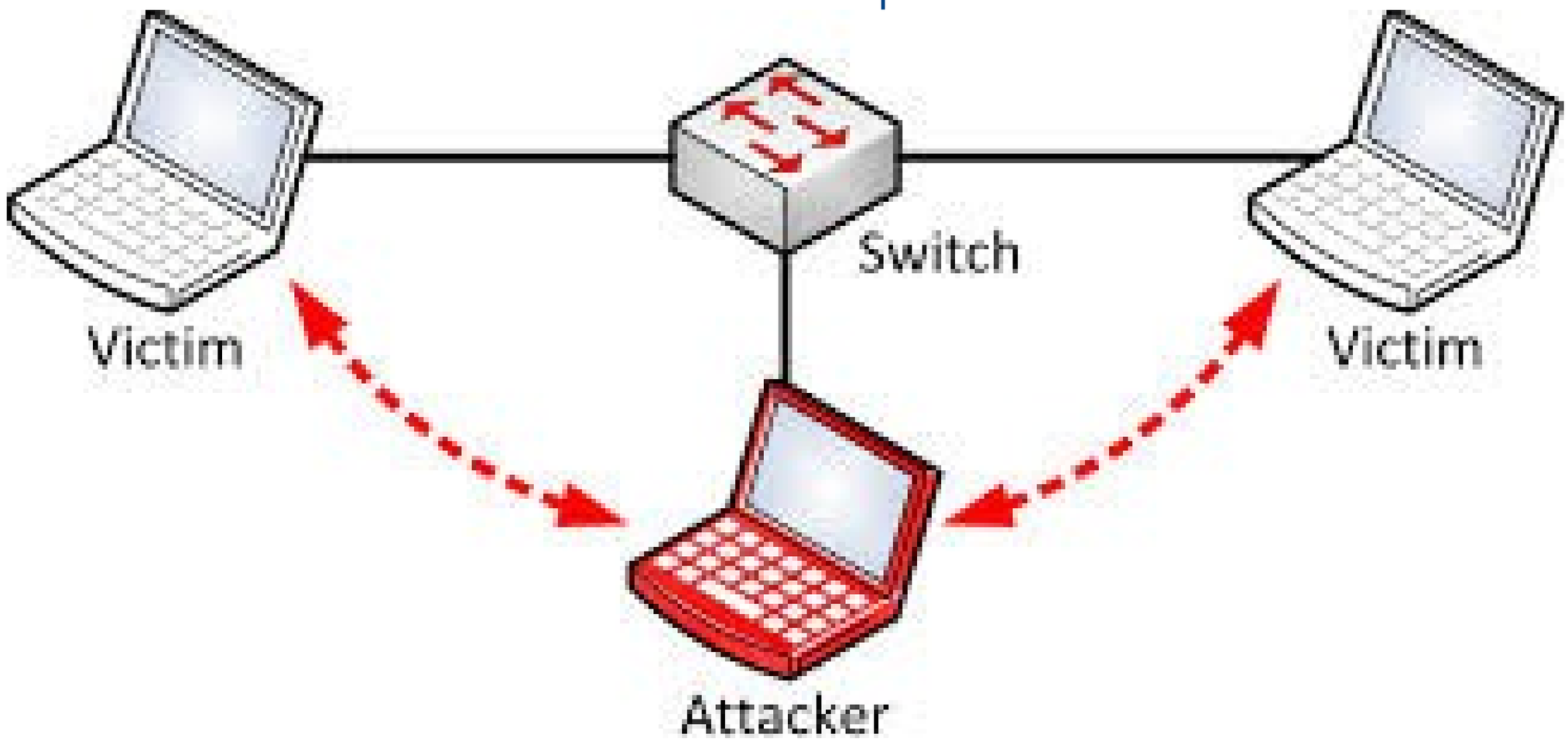


Neighbor Solicitation

Neighbor Advertisement

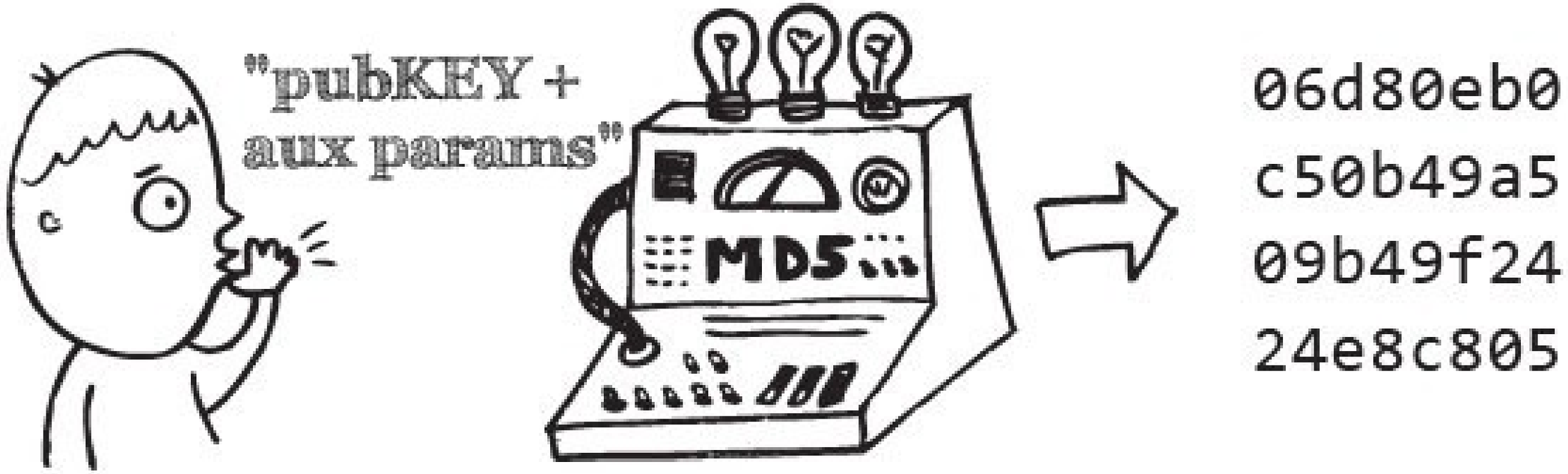
NDP Vulnerability

- Just like in ARP, NDP is also vulnerable to attacks.
- NDP's Neighbor Solicitation / Advertisement can be spoofed.



SEND

- Send uses the cryptographic hash of a public key and auxiliary parameters to generate CGAs.



Process

- Edit source code of ndisc.c to include a new struct for neighbor advertisement and neighbor solicitation messages.
- Compile and build stable kernel (with new features)

ndisc.c

```
1027            /* XXX: idev->cnf.proxy_ndp */
1028            goto out;
1029        }
1030
1031 /* CynthiaO
1032     declare struct for neighbor advertisement message */
1033     ndsec_ndiscmsg NAMsg;
1034
1035     memcpy(NAMsg.macAddress, lladdr, 6);
1036     NAMsg.targetIP =        *daddr;
1037     NAMsg.sourceIP =        *saddr;
1038     NAMsg.opcode =          NEIGH_OPCODE_ADVERTISEMENT;
1039
1040
1041 // relay neighbor advertisement
1042 if (ndsec_rchan) {
1043                 /* Pop up this msg into user space */
1044                 ndsec_msg_counter++;
1045                 ndsec_rlmsg rlmsg;
1046
1047         /* Copy the ND msg */
1048                 memcpy(&(rlmsg.ndsec_nd_msg), &(NAMsg), sizeof(ndsec_ndiscmsg));
1049
1050         /* Save the net_device ptr */
1051         rlmsg.ndsec_dev_ptr = dev;
1052
1053         /* Send the rlmsg via relay */
1054                 relay_write(ndsec_rchan, &rlmsg, sizeof(rlmsg));
1055                 printk(KERN_INFO "ndsec: relay written for msg %lu in nd_process()\n",  ndsec_msg_counter);
1056                 goto out;
1057         } else {
1058                 printk(KERN_INFO "ndsec: relay not ready in nd_process()\n");
1059         }
1060
1061 /* Comment out neighbor discovery update
1062         neigh_update(neigh, lladdr,
1063                      msg->icmph.icmp6_solicited ? NUD_REACHABLE : NUD_STALE,
```

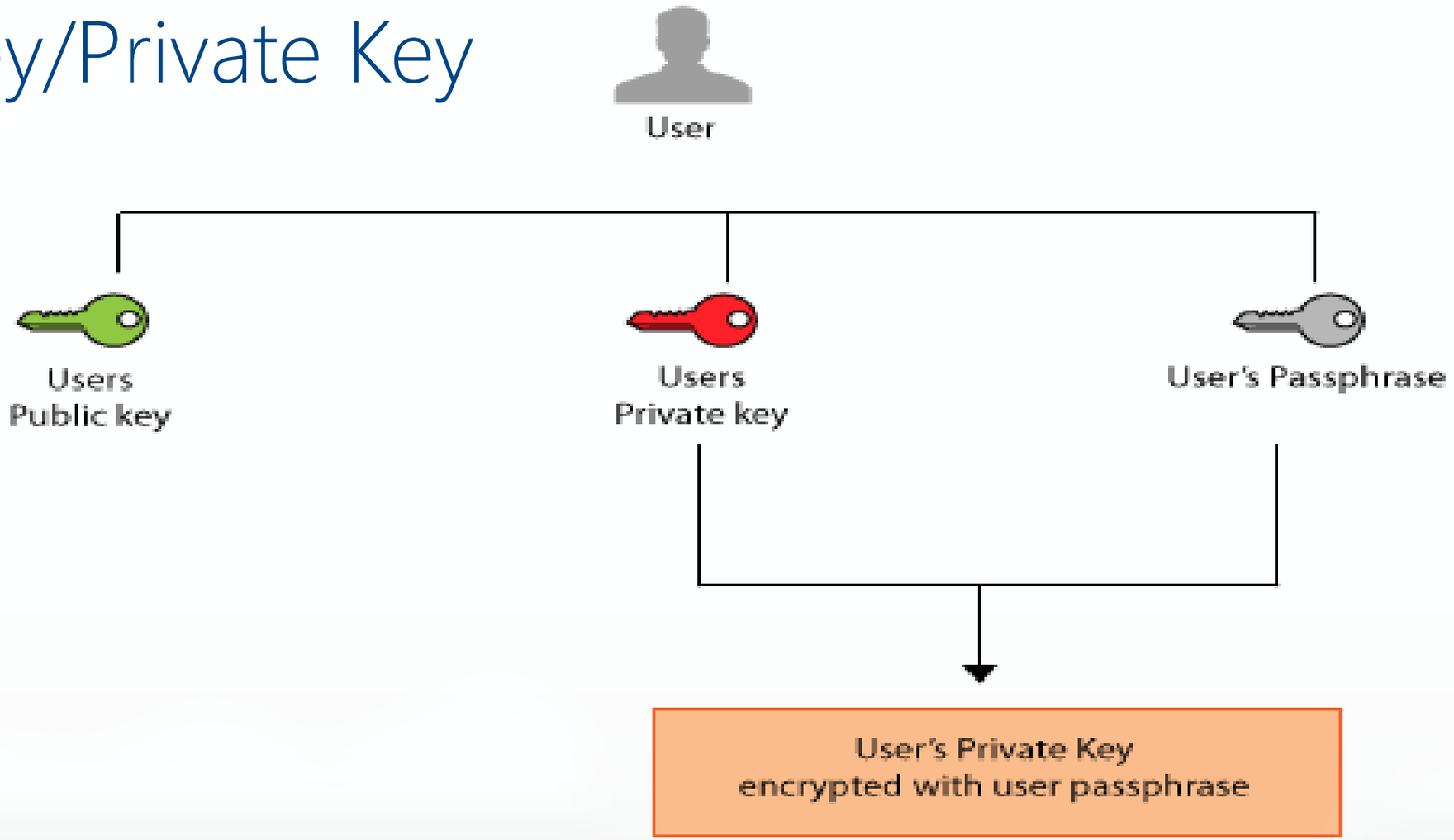
NDprotector

- When a Neighbor Discovery message (ICMPv6 packet) is received or emitted by an interface, a hook set by ip6tables redirect the packet to the userspace.

- This extraction is performed by the libnetfilter_queue.

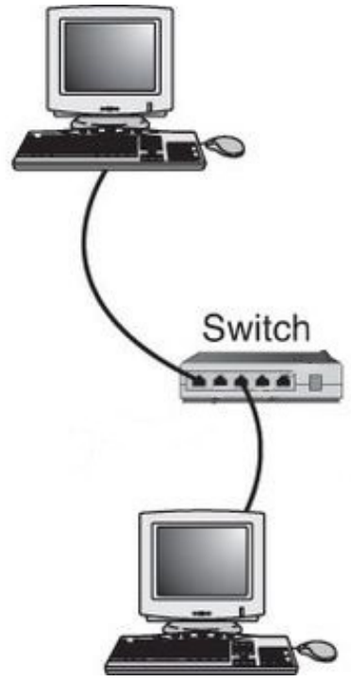
NDprotector

- Scapy6 dissects each intercepted messages
- Each assigned address is bound to a Public Key/Private Key



Process cont.

- Create a local ipv6 network



- Install Ndprotector

- Generate public and private keys using openssl

- Edit host configuration file to replace key paths

Process cont.

- Test ping6 with Ndprotector running
- Use wireshark, to determine message type

Wireshark

2001:db8:0:100::80	ff02::1:ff00:79	ICMPv6	86 Neighbor Solicitation for 2001:db8:0:100::79 from 0
2001:db8:0:100::79	2001:db8:0:100::80	ICMPv6	86 Neighbor Advertisement 2001:db8:0:100::79 (sol, ovr
2001:db8:0:100::80	2001:db8:0:100::79	ICMPv6	118 Echo (ping) request id=0x0ce1, seq=1, hop limit=0 (
2001:db8:0:100::79	2001:db8:0:100::80	ICMPv6	118 Echo (ping) reply id=0x0ce1, seq=1, hop limit=0 (re

```
▼ICMPv6 Option (CGA)
   Type: CGA (11)
   Length: 41 (328 bytes)
   Pad Length: 5
   Reserved
  ▶CGA: 0000000000000000000000000003220010db800000100...
   Padding
▶ICMPv6 Option (Timestamp)
▼ICMPv6 Option (Unknown 42)
   Type: Unknown (42)
   Length: 2 (16 bytes)
  ▶[Expert Info (Note/Undecoded): Dissector for ICMPv6 Option (42) code not implemented, Contact Wireshark developers if you want this supported]
   Data: 070081800a0b09000000000000000c23
▼ICMPv6 Option (RSA Signature)
   Type: RSA Signature (12)
   Length: 35 (280 bytes)
   Reserved
   Key Hash: 3a61f33898601c51fac62de5c8012eca
   Digital Signature and Padding
```

What's Next?

- Continue to debug Ndprotector
- Work on ARPsec daemon to implement IPv6 functionality without messing up the already configured Ipv4 implementation