

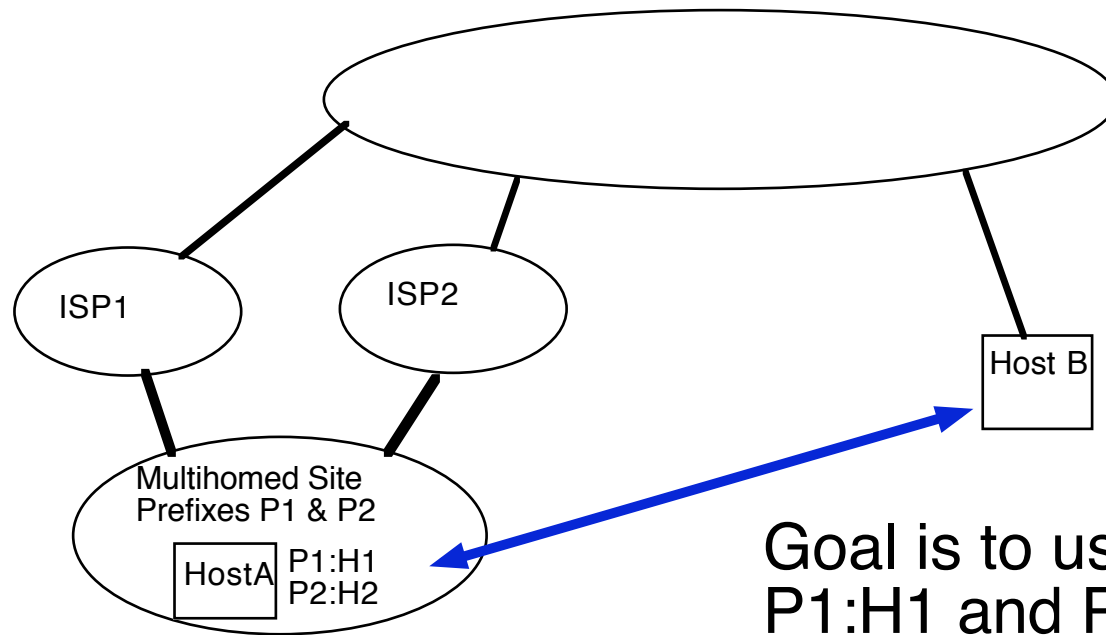
Hash Based Addresses

HBA

multi6 wg meeting - IETF 61

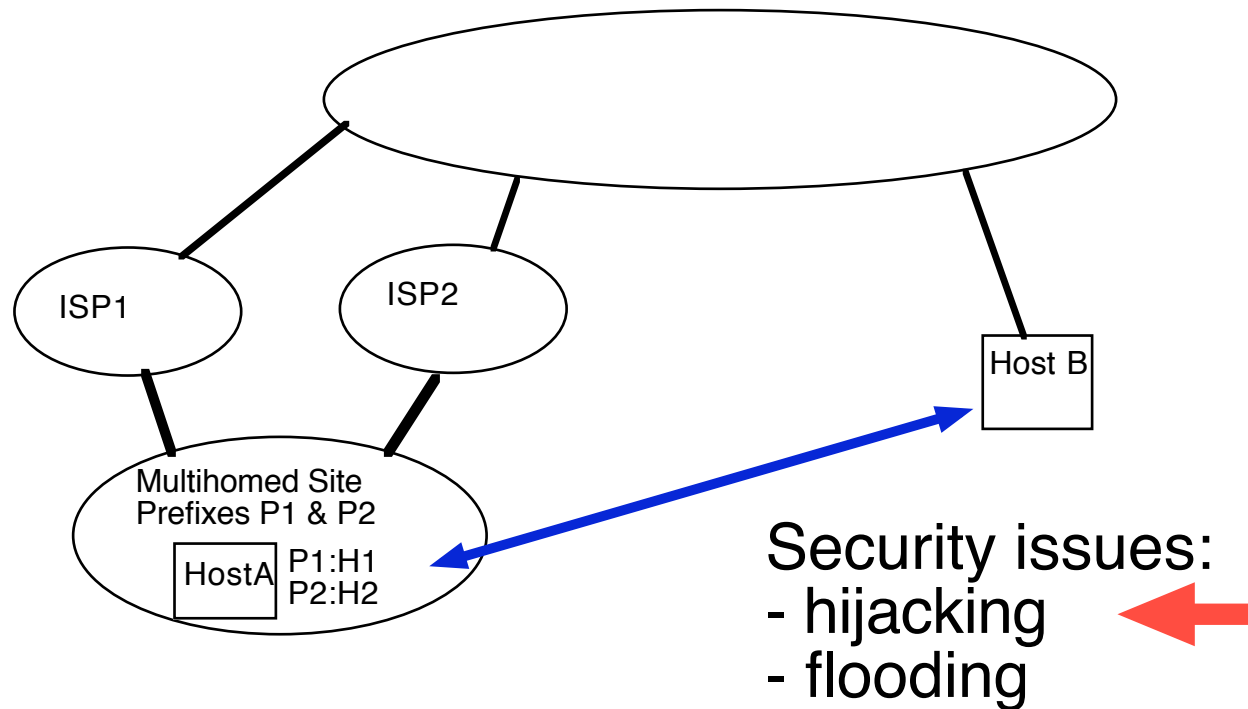
MULTI6 dt: J. Arkko, M. Bagnulo,
I. van Beijnum, G. Huston, E.
Nordmark, M. Wasserman, J. Ylitalo.

Multihoming scenario

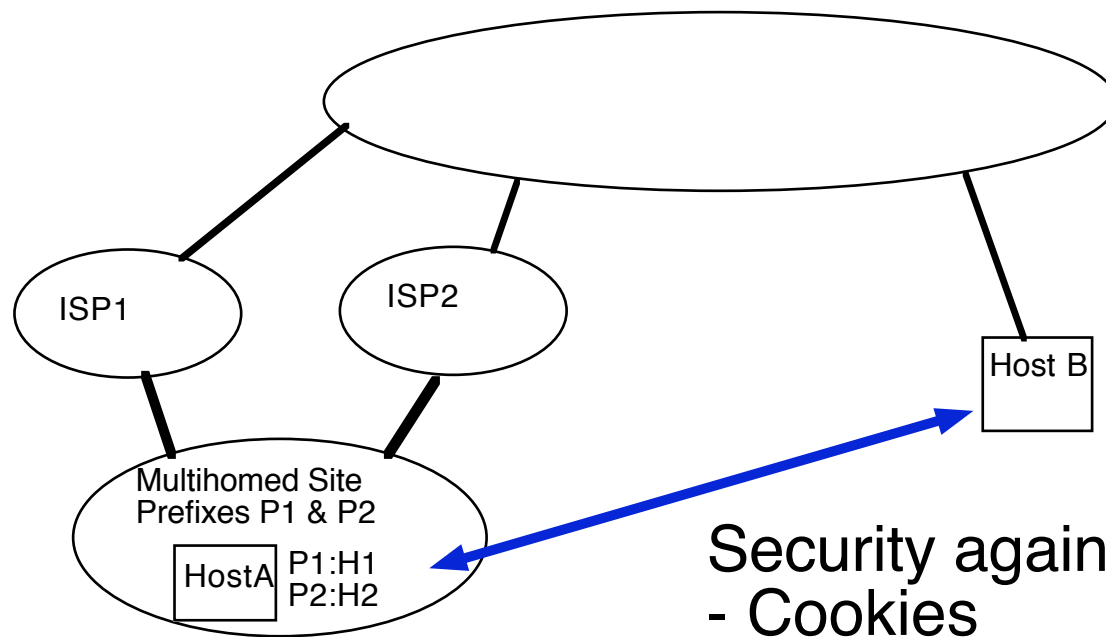


Goal is to use addresses P1:H1 and P2:H2 in the same communication when needed (e.g.outage)

Multihoming scenario



Multihoming scenario



Security against hijacking:

- Cookies
 - time-shifting attacks
- Public key crypto
 - overkill for general case

Characteristics of HBAs

- Generate sets of securely bound addresses
 - Not vulnerable to time-shifted attacks
 - Not using public key crypto
- The resulting set is static
 - i.e. no addition of new addresses once the set is created

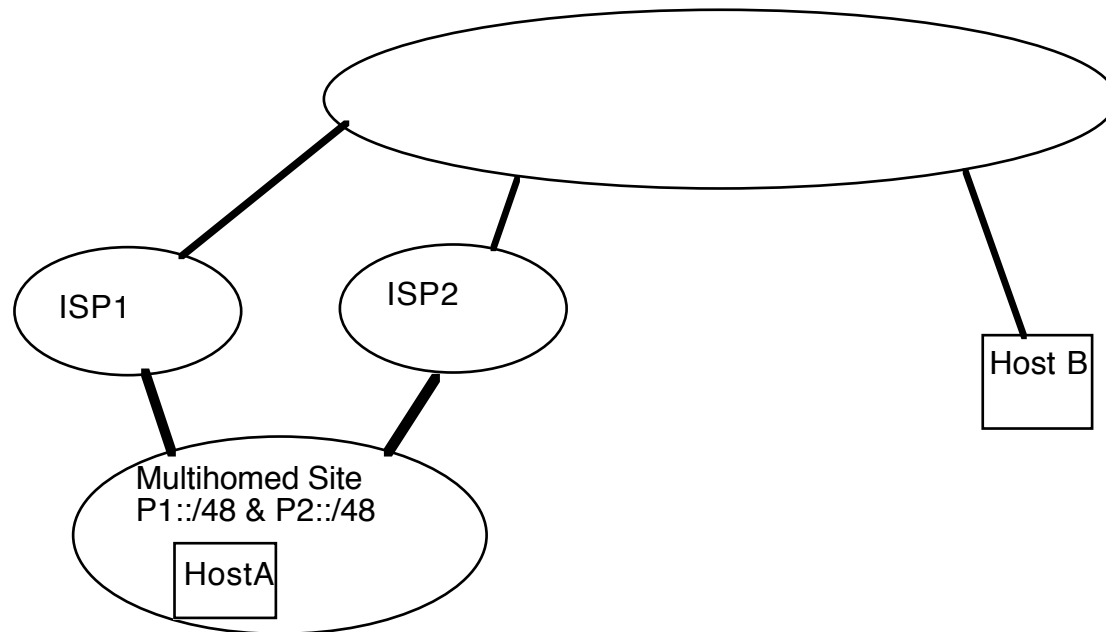
Main idea of HBAs

- Include the information about multiple prefixes in the addresses themselves
- Available prefixes: P_1, P_2, \dots, P_n
- Generate iid as:
 $iid = \text{Hash}(P_1 | P_2 | \dots | P_n | \text{rand})$
- Resulting HBA is Pref:iid being
Pref= P_i for $i=1, \dots, n$
- (Privacy is considered later...)

Application to multihoming

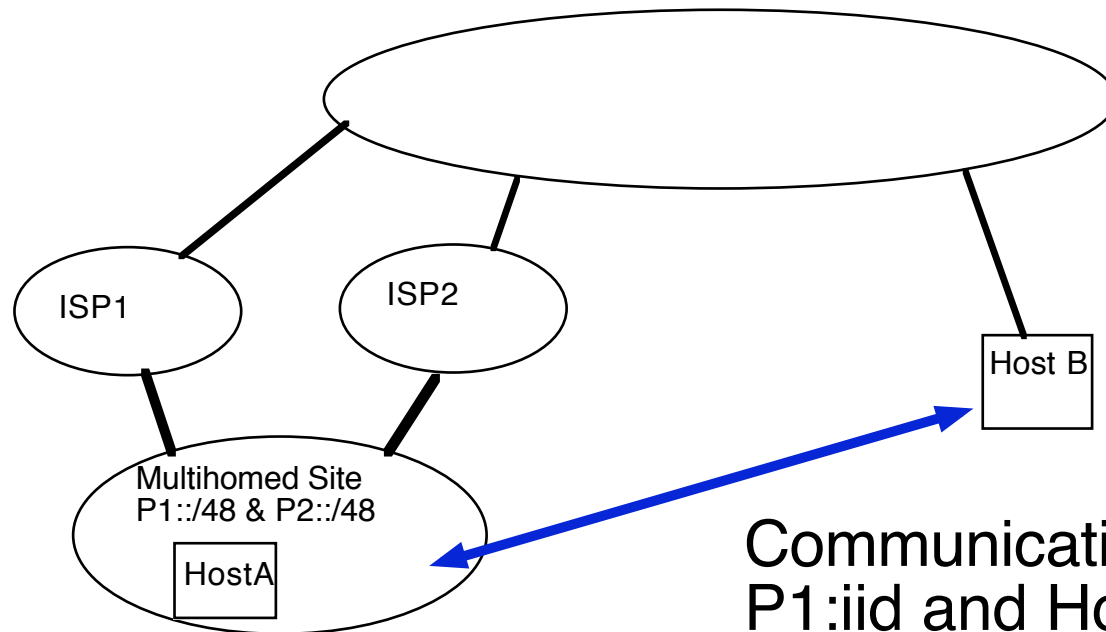
- HostA in multihomed site: P1 & P2
- Generate HBAs for HostA
 - $iid = \text{Hash}(P1 | P2 | \text{rand})$
 - Addresses for HostA: P1:iid and P2:iid
- Then, HostA communicates with HostB using P1:iid
- HostA informs P1, P2 & rand to HostB
- An outage forces to use P2
- HostB can verify that P2:iid is valid for HostA using HBA

Application to Multihoming



$iid = \text{Hash}(P1|P2|rand)$
Addresses for HostA
P1:iid
P2:iid

Application to Multihoming



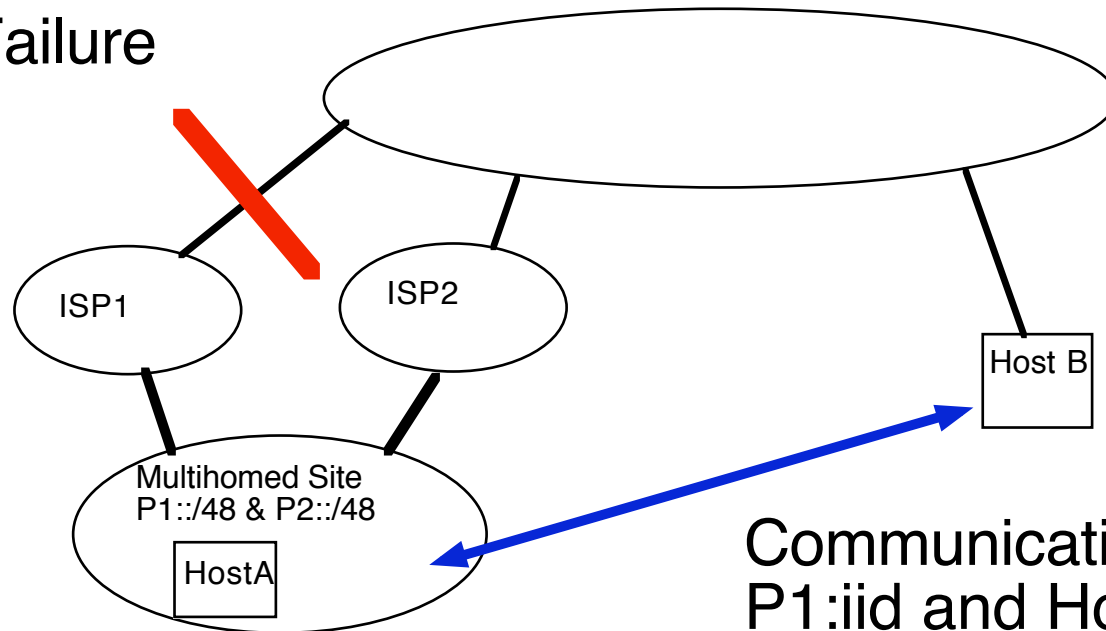
$iid = \text{Hash}(P1|P2|rand)$
Addresses for HostA
P1:iid
P2:iid

Communication using
P1:iid and HostB as
ULPIDs and locators

(HostA sends rand & set of
prefixes used in HBA)

Application to Multihoming

Failure

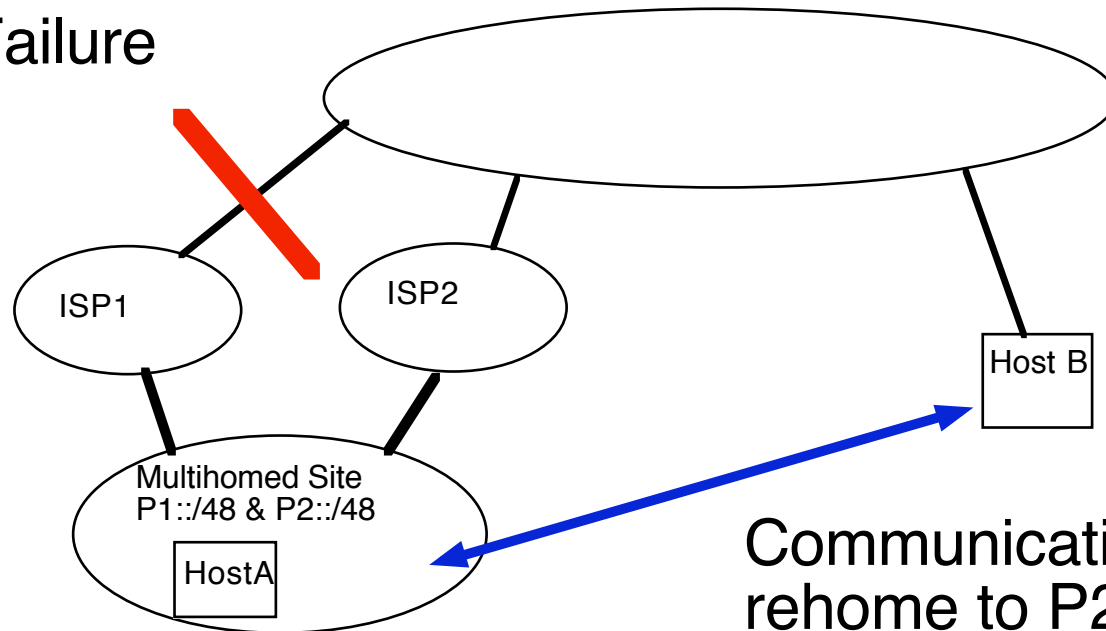


$iid = \text{Hash}(P1|P2|rand)$
Addresses for HostA
P1:iid
P2:iid

Communication using
P1:iid and HostB as
ULPIDs and locators

Application to Multihoming

Failure



$iid = \text{Hash}(P1|P2|rand)$
Addresses for HostA
P1:iid
P2:iid

Communication must
rehome to P2:iid
HostB verifies that P2 is
included in the prefixes
used to generate iid

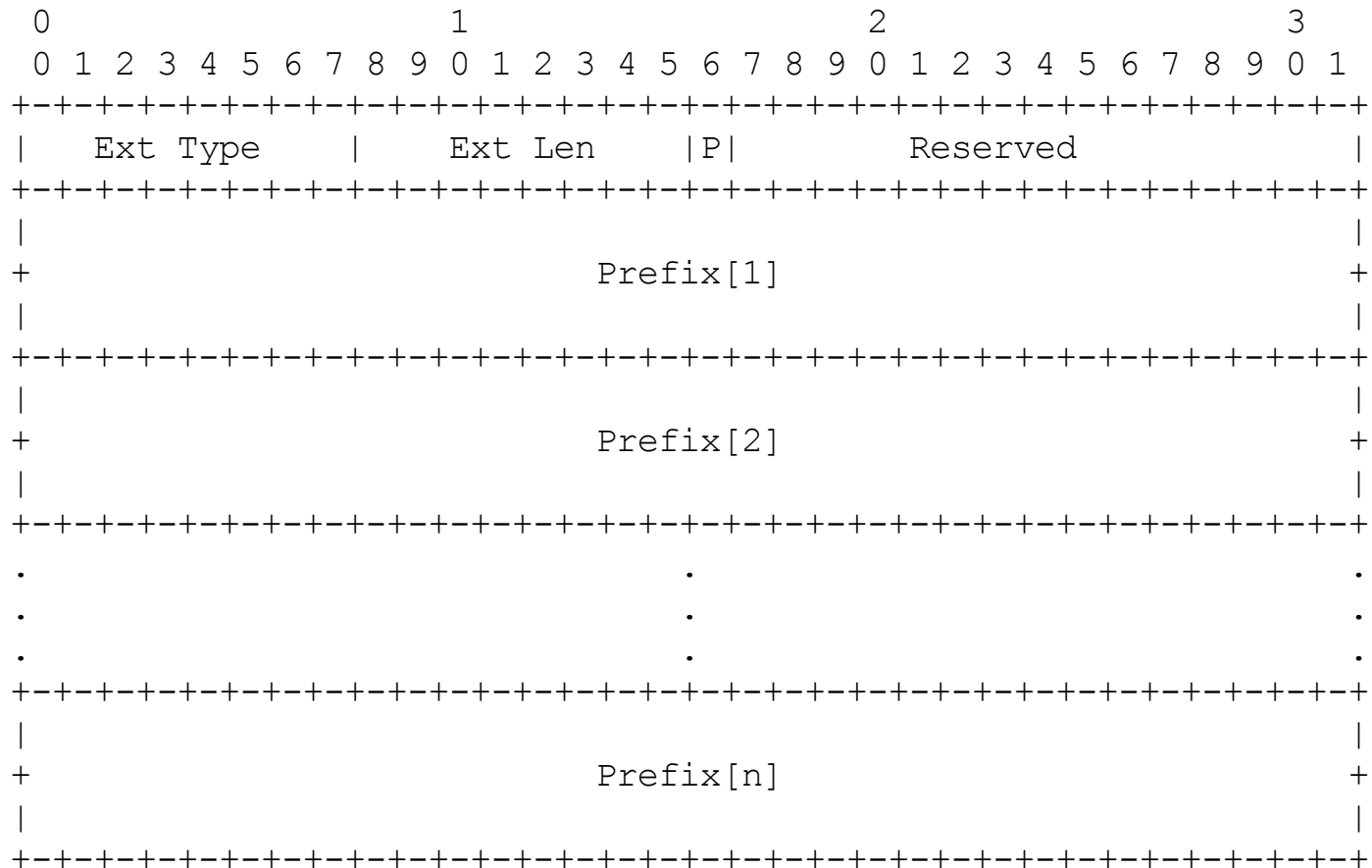
Compatibility with CGAs

- HBAs and CGAs use the iid bits
- Reasons for compatibility
 - SeND uses CGAs, so compatibility is required to use SeND and HBA-based multi6 simultaneously
 - A CGA-based multi6 supports dynamic prefix set, which is useful in some scenarios e.g. mobility, renumbering
- Result: define HBA as a CGA extension

Resulting address types

- CGA-only addresses:
 - the hash contains public key
- CGA/HBA addresses:
 - the hash contains public key and prefix set
- HBA-only addresses
 - the hash contains prefix set only

CGA Multi-Prefix Extension



HBA-set generation

- Inputs
 - A vector of n 64-bit prefixes
 - A Sec parameter,
 - A public key (in the case of CGA/HBA)
- Outputs
 - An HBA-set
 - their respective CGA Parameters
Data Structures

HBA-set generation

1. Multiprefix extension generation
2. Random Modifier generation
 1. if HBA only, then Ext mod instead of pk
3. Hash2 generation
 1. Modifier|0s|pk*|multipref
4. Verify $16 * \text{Sec lsbits}(\text{Hash2}) = 0$
5. HBA-set generation. For $i=1$ to n do:
 1. Generate Hash1[i] (Modifier|Pi|cc|pk*|mp)
 2. Generate HBA[i] (Pi:64lsb Hash1 (u,g,sec))
 3. Generate CGA PDS[i]

HBA-set verification

- Inputs
 - An HBA
 - A CGA Parameter Data Structure
- Verification process
 - Verify that the 64-bit HBA prefix is included in the Multi-Prefix Extension.
 - Run the CGA verification process (including the Multi-prefix extension in the verification process)

Security considerations

- Basic attack to HBA:
 - Given an HBA set P1:H1, P2:H2
 - Generate CGA_PDS such as
 - P1 and PX is included
 - the resulting iid==H1
 - so that, the attacker can redirect communications established with P1:H1 to PX:HX
 - The difficulty is to find a suitable modifier i.e. brute force attack
 - The difficulty is $O(2^{59})$
 - Sec>0 then is $O(2^{(59+16 * Sec)})$

Privacy considerations

- CGA_PDS[i] contains P_i in the subnet field of basic CGA PDS
- Hence the iid of the HBAs of a given HBA set will be different

Feedback so far...

- First implementation by Francis Dupont (ENST)
- Some additional comments on the ml