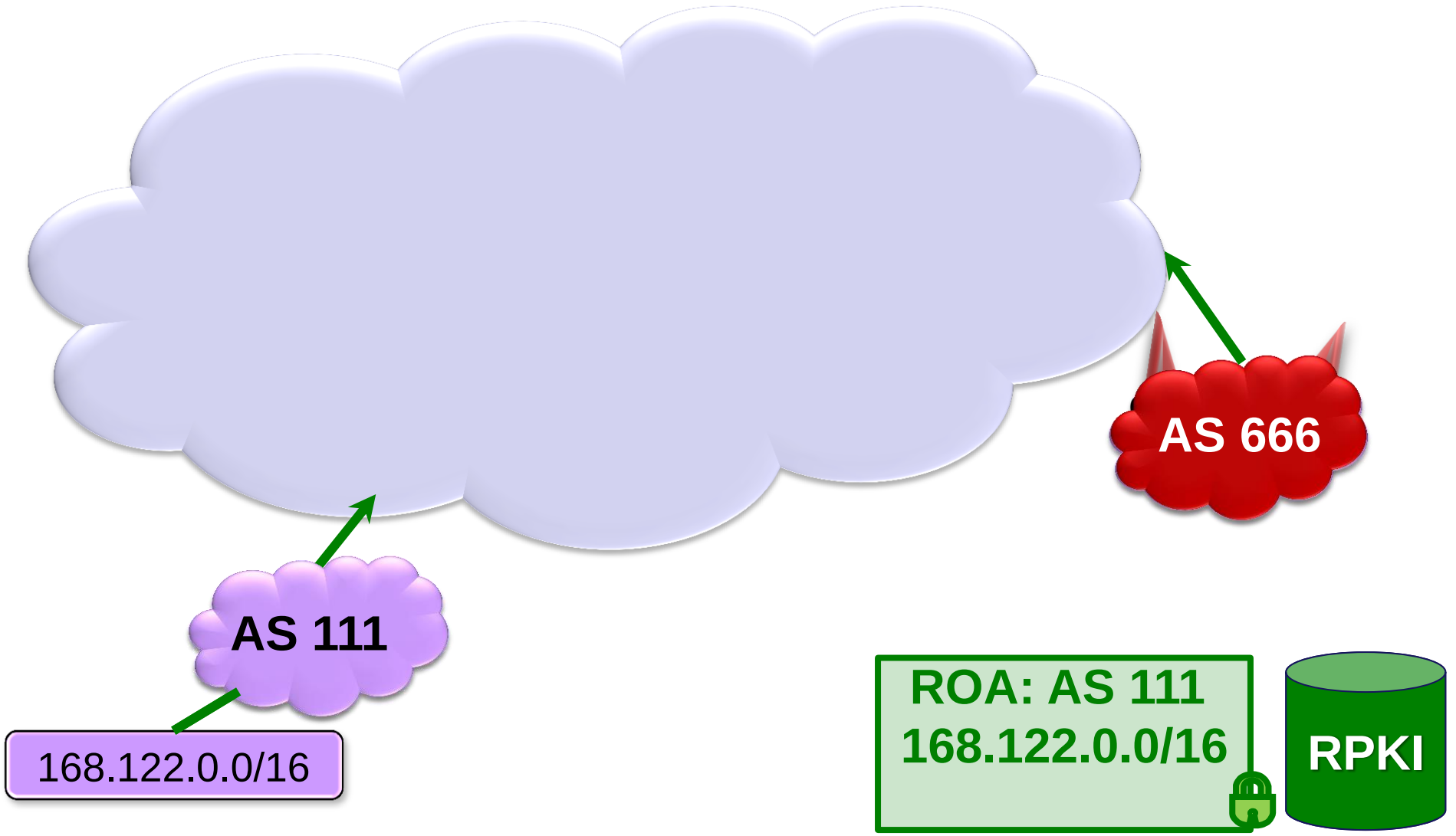
sidrops@IETF'98
Chicago, March 2017

draft-yossigi-rpkimaxlen-00

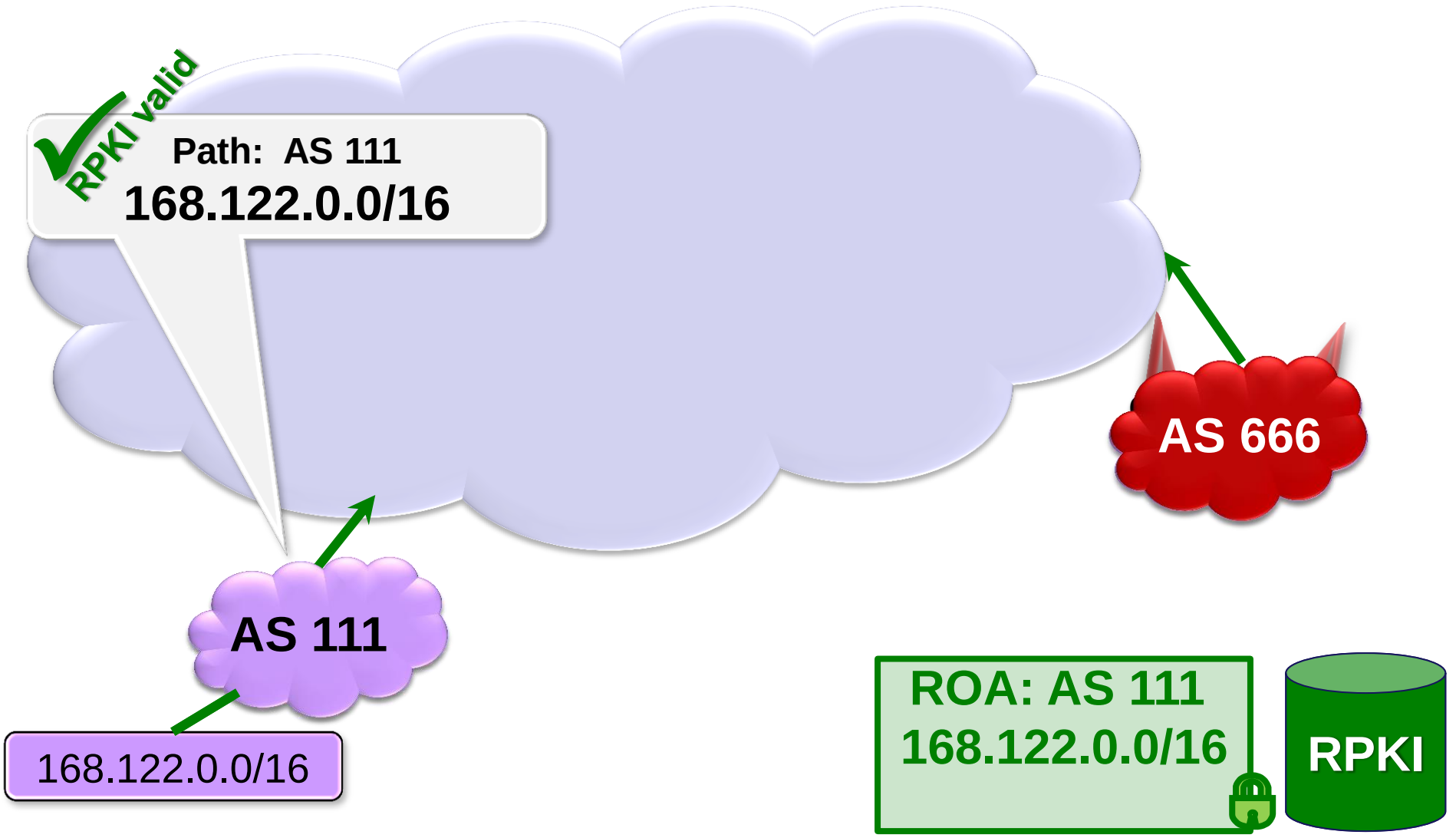
The use of maxLength in the RPKI

Yossi Gilad (Boston University)
Sharon Goldberg (Boston University)
Kotikalapudi Sriram (NIST)

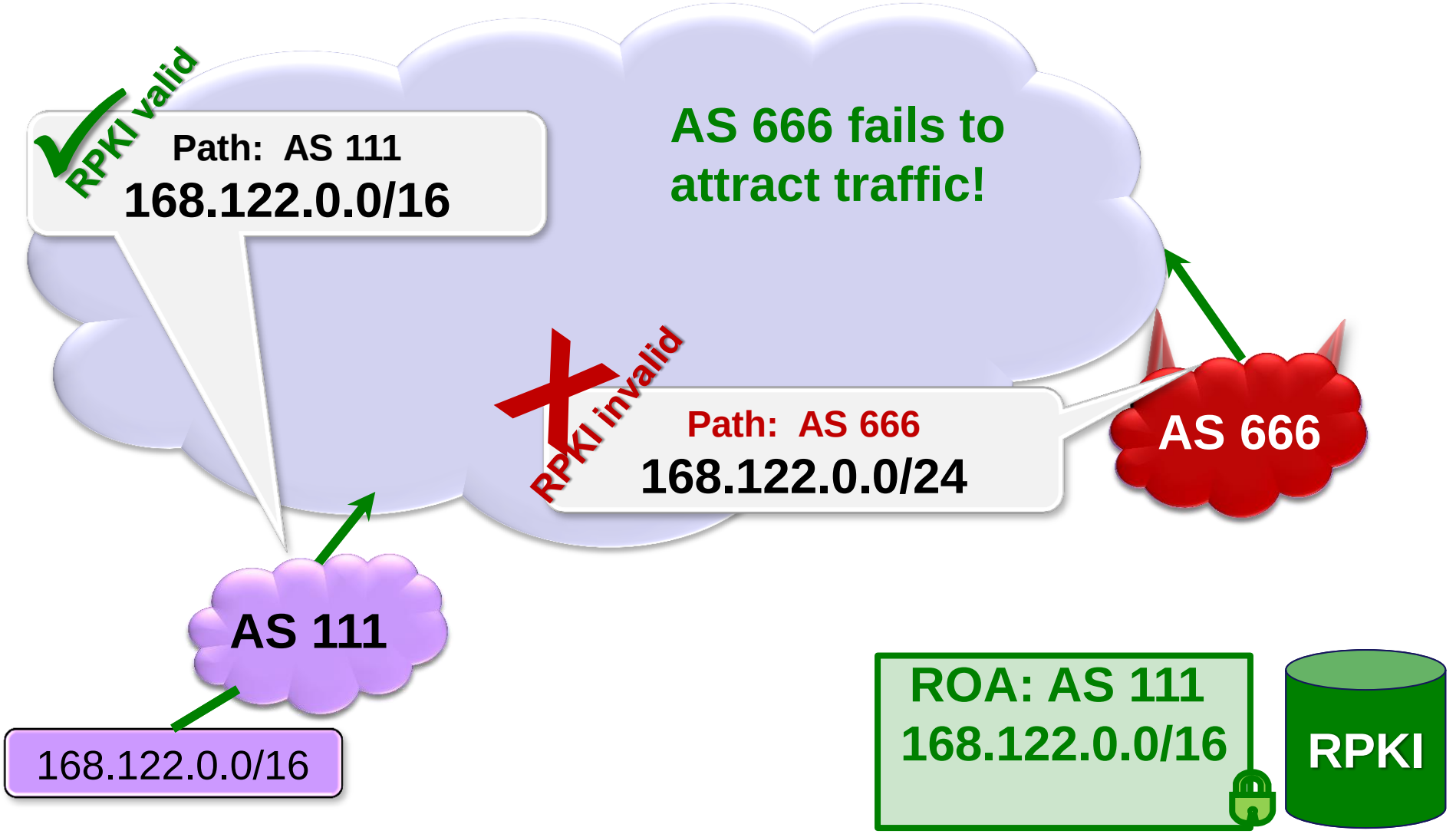
When used properly, the RPKI defeats subprefix hijacks



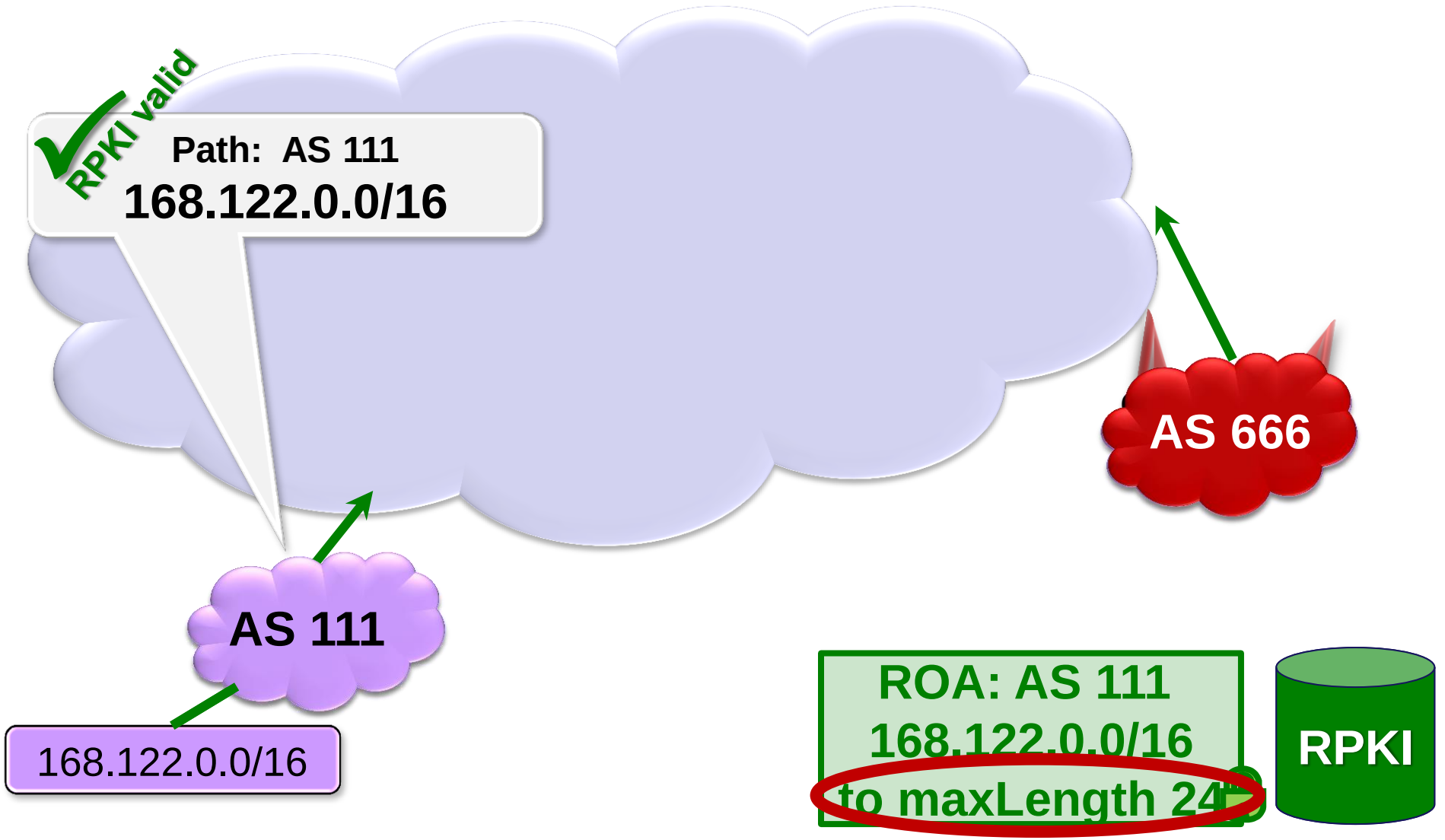
When used properly, the RPKI defeats subprefix hijacks



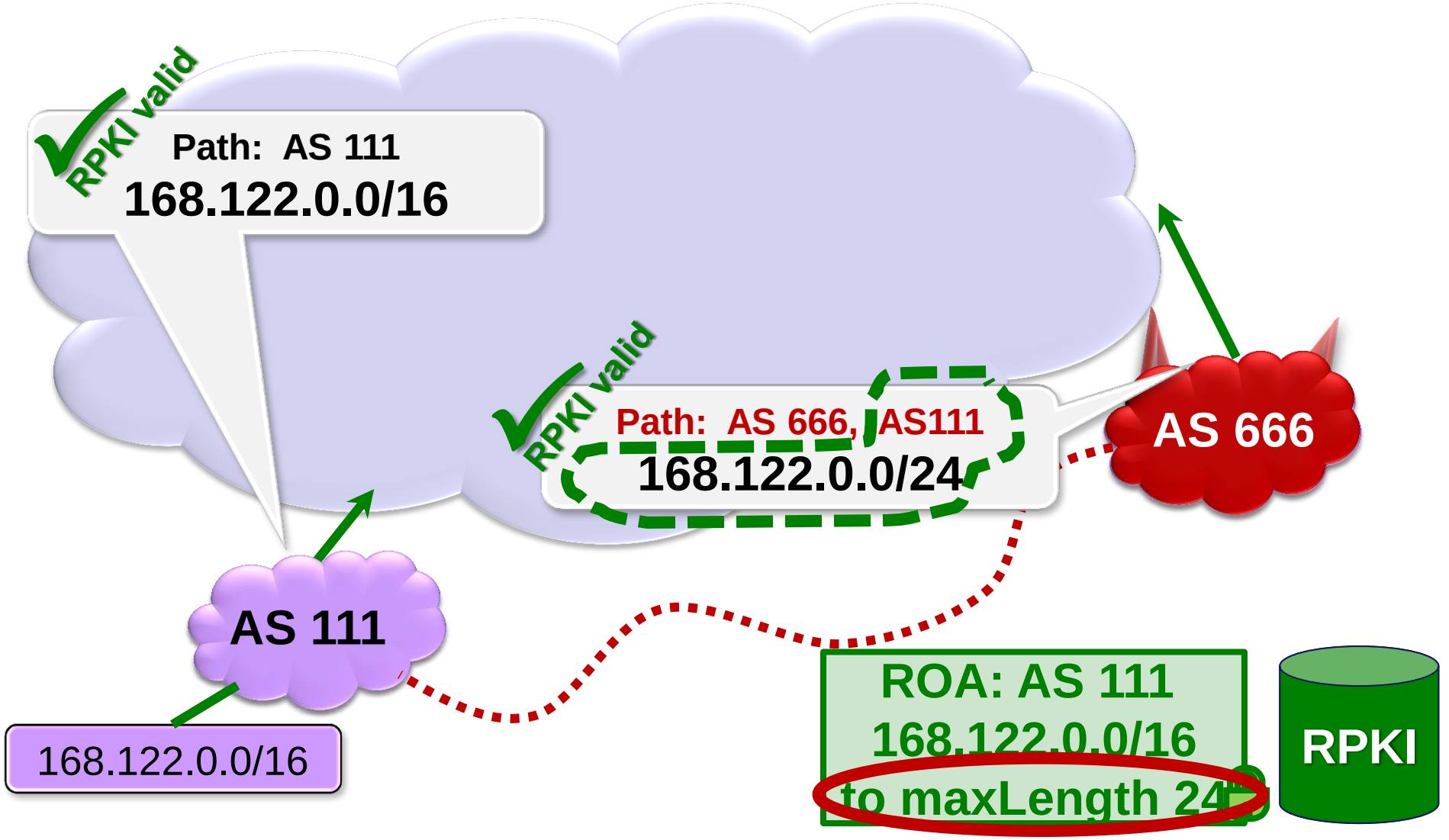
When used properly, the RPKI defeats subprefix hijacks



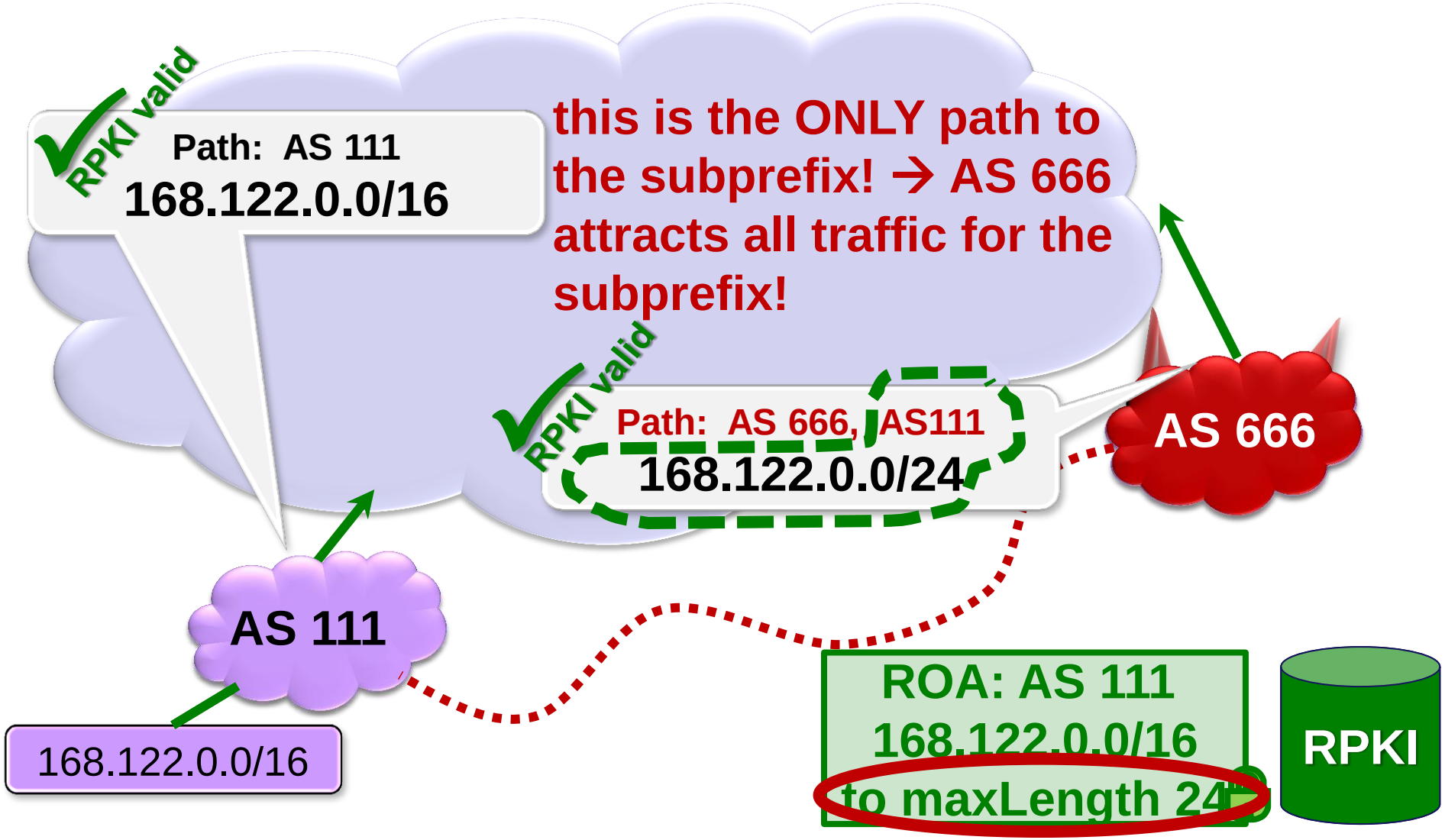
Loose maxlength → forged-origin subprefix hijack



Loose maxlength → forged-origin subprefix hijack

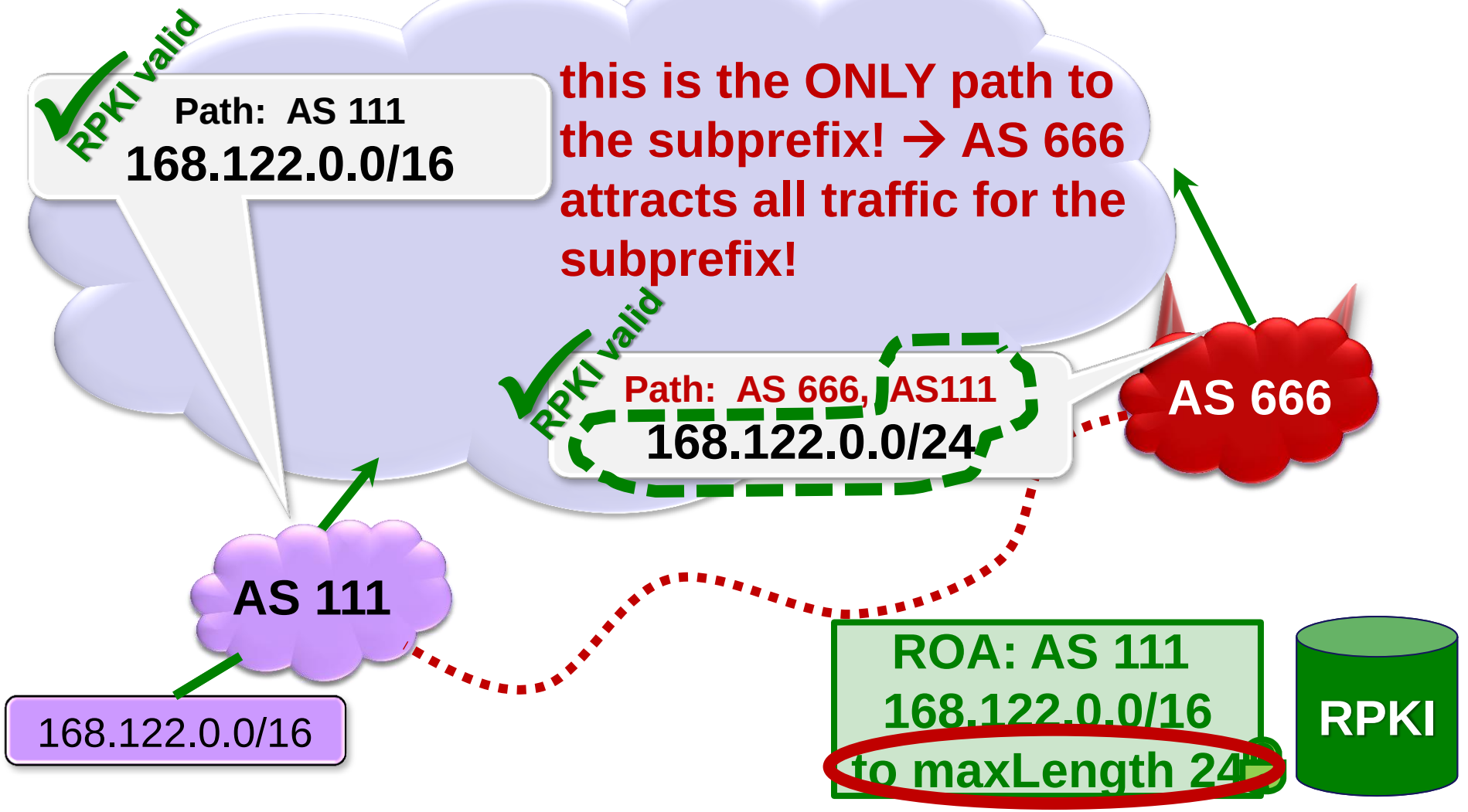


Loose maxlength → forged-origin subprefix hijack



Loose maxlength → forged-origin subprefix hijack

attack is highly effective because 168.122.0.0/24 is unannounced



maxLength misconfigurations are common!

- forged-origin subprefix hijack affects any prefix in ROA where
  - maxlength **m** > prefixlen **p**, unless
  - every subprefix of length **m** is announced in BGP

maxLength misconfigurations are common!

- forged-origin subprefix hijack affects any prefix in ROA where
  - maxlength **m** > prefixlen **p**, unless
  - every subprefix of length **m** is announced in BGP

- 16% of the IP prefixes in ROAs have maxlength > prefixlen
- 89% of these are vulnerable to forged-origin subprefix hijacks
  - Even large providers are vulnerable

Recommendations

- As a best common practice:
  - Operators should refrain from using maxlength in ROAs. UIs should convey that.
  - ROAs should instead have explicit **lists** of prefixes authorized to be originated by a single AS
  - Whenever possible, use **minimal** ROAs where each listed prefix is originated in BGP.
- The RPKI already supports this. No extra ROAs needed.

Recommendations

- To reduce the number of RPKI filtering rules, we developed software that RPKI local caches can use to compress lists of prefixes from ROAs back to (AS, prefix, maxlength) tuples

  https://github.com/yossigi/compress_roas

- See our technical report: http://eprint.iacr.org/2016/1015.pdf