Discussion of Key Rollover Mechanisms for Replay-Attack Protection

November 9, 2012

IETF-85 SIDR WG Meeting

Kotikalapudi Sriram and Doug Montgomery

NIST

Contact: ksriram@nist.gov

Purpose of the Document

- Intended to be a design discussion document complementing draft-ietf-sidr-bgpsec-rollover
- Provides taxonomy, descriptions of various key rollover alternatives, and captures many of the discussions that have occured in SIDR about the replay-attack protection

Key Rollover (KR) Method

- Key Rollover (KR) method has different flavors as explained in the slides that follow
- The following features are common to all KR methods
- In the KR method, it is best if the BGPSEC router has two pairs of certs as follows:
  - A pair of origination certs (current and next) for use with prefixes being originated by the AS of the router; and
  - A pair of transit certs (current and next) for use with transit prefixes.
- Note: If a BGPSEC router only originates prefixes (i.e., has no transit prefixes), then it needs to maintain only a pair of origination certs
- Three KR methods that are described in the slides that follow differ in how the rollover of certs (or keys) is done

Periodic Key Rollover (PKR)

- Router's origination cert's NotValidAfter time is used as the implicit expire time for origin's signature
- "Beaconing" is periodic re-origination of prefixes by origin ASes
- "Beacon" before NotValidAfter time of the Current cert
- At beacon time, Next cert becomes new Current cert, and a New "next" cert is created and propagated
- Distributed actions by prefix owners
- Transit cert can have a very large NotValidAfter time (say ~years)
- Big upside: Less load on transit routers (no need to re-propagate all transit prefixes when peering or policy changes occur)
- Downside: Some churn in BGPSEC and RPKI; Every BGPSEC router rolls origination cert (key) once every "beacon" interval

Event-driven Key Rollover (EKR)

- Key rollover is reactive to events (not periodic)
- If a peering change event involves only prefixes being originated at this AS, then the router rolls only the origination key
- If a peering change event involves transit prefixes at this AS, then the router rolls the transit key as well as the origination key
- If a key rollover takes place, then a corresponding (origination or transit) new “next” cert is propagated in RPKI
- Big upside (relative to PKR): No churn in BGPSEC and RPKI as long as no triggering events occur
- Big downside (relative to PKR): Whenever the transit key is rolled, there is a storm of BGPSEC updates at routers in large transit ASes! Comment:
  - But it can be flow controlled / jittered. The added convergence time may not be damaging because the data packet delivery is not impacted? Needs measurement/modeling.

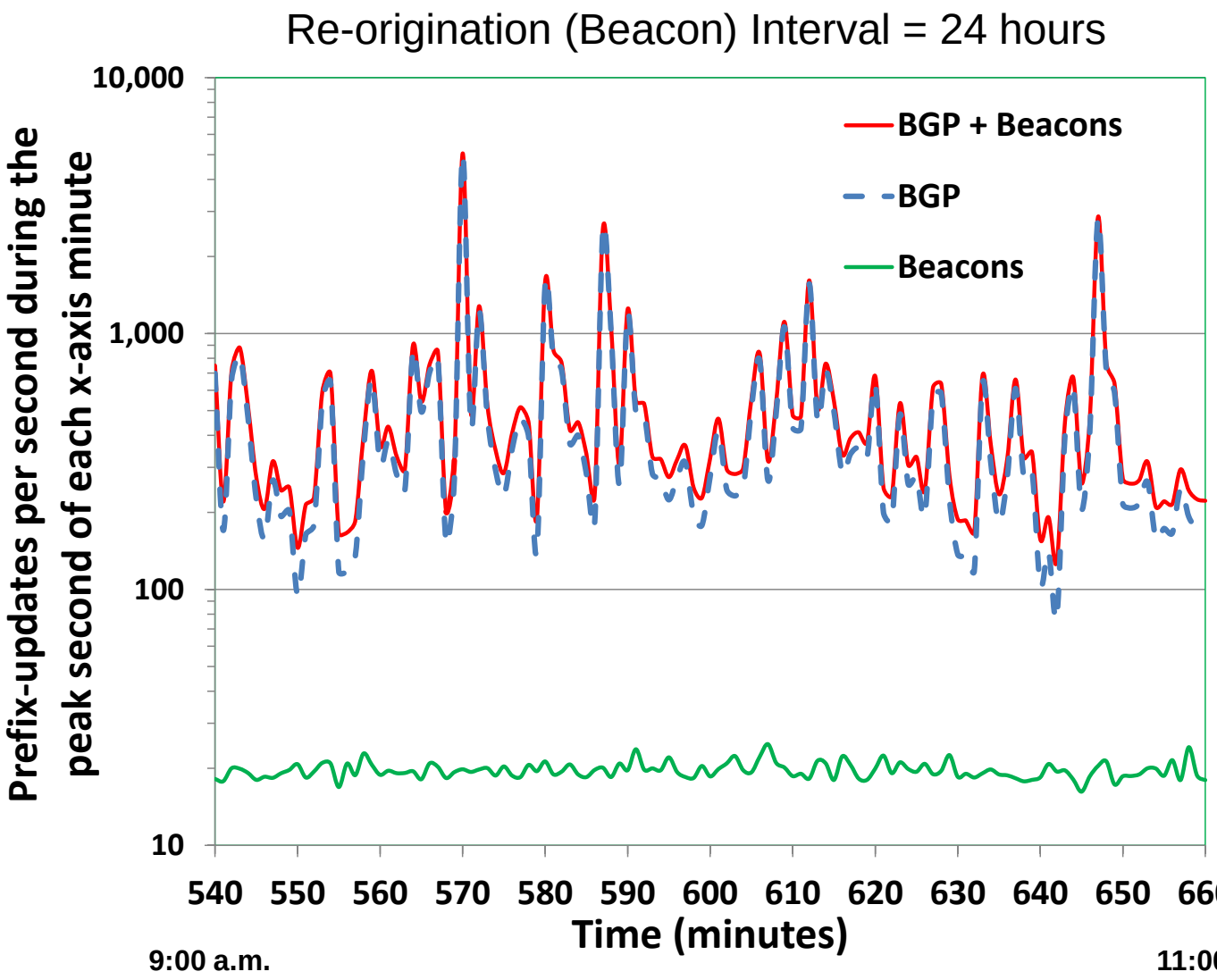
EKR-A: EKR where Update Expiry is Enforced by CRL

- NotValidAfter time of origination and transit certs is set to a large value (~years)
- Whenever key roll (for origination or transit) occurs, then CRL is propagated for the old cert
- So the old update expires (due to invalid state) only when the CRL propagates and reaches the reliant router
  - RPKI cache server sends withdraw of corresponding Pub Key to reliant router
- Downside: Router needs to receive CRL (or Pub-Key withdrawal from RPKI cache) in order to know update has expired; This is CRL propagation time dependent

EKR-B: EKR where Update Expiry is Enforced by NotValidAfter Time

- NotValidAfter time of current origination and transit certs is set to a value determined by desired vulnerability window (~day)
- Update expiry is controlled by NotValidAfter time and CRL is not sent for the old cert when key rollover happens
- If no triggering event occurs to cause origination key roll within a pre-set time (< NotValidAfter), then new origination cert is issued only to extend the NotValidAfter time but the corresponding key pair and SKI remain unchanged.
- Likewise for the transit (current and next) certs/keys
- Upside: Routers do not get any RPKI updates from the cache server when cert changes but key pair and SKI remain unchanged
  - Routers do not receive NotValidAfter time
  - RPKI cache keeps track of NotValidAfter time
  - RPKI cache provides to router only valid {AS, SKI, Pub Key} tuples

Load Due to BGP and BGPSEC/PKR Periodic Re-Originations (i.e. Beacons) for 3 Peers



Using Routeviews data, Feb 1, 2012.

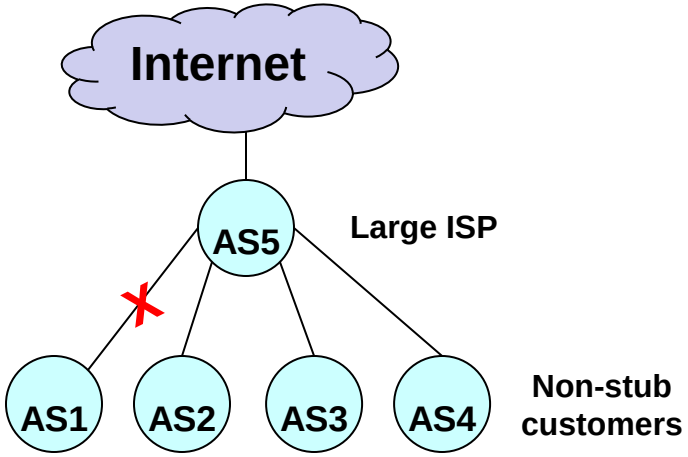
BGP feeds from AS7018, AS 701, and AS 3356 peer routers combined.

BGPSEC/PKR router in consideration receives full tables from three peers in AS7018, AS 701, AS 3356.

Update load due to beacons in PKR method is estimated using a Poisson model.

Comparison of PKR vs. EKR: Scenario 1

Peering Change Event Scenario 1:



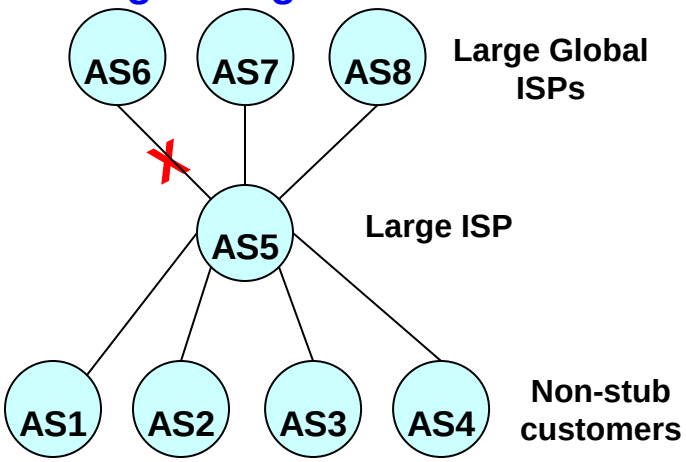
- Assume each AS in this figure also represents a single BGPSEC router
- We focus on workload at the router in AS5
- AS1 thru AS4 are non-stub customers of AS5; Each receives almost full table (400K signed prefix updates) from AS5
- Assume: AS1 and its customers together originate 100 prefixes total; likewise for AS2, AS3, AS4
- Event: Peering between AS1 and AS5 is discontinued

Workload Comparison:

- When the peering (AS5-AS1) is discontinued:
  - ❖ In the PKR method, the router at AS5 sends only 4x100 = 400 Withdraws in total and signs/re-propagates ZERO prefix updates
  - ❖ In contrast, in the EKR method (EKR-A or EKR-B), the router at AS5 sends those same 400 Withdraws but also signs and re-propagates 3x400K +3*200 +300 = 1.2 MILLION signed prefix updates in total

Comparison of PKR vs. EKR: Scenario 2

Peering Change Event Scenario 2:



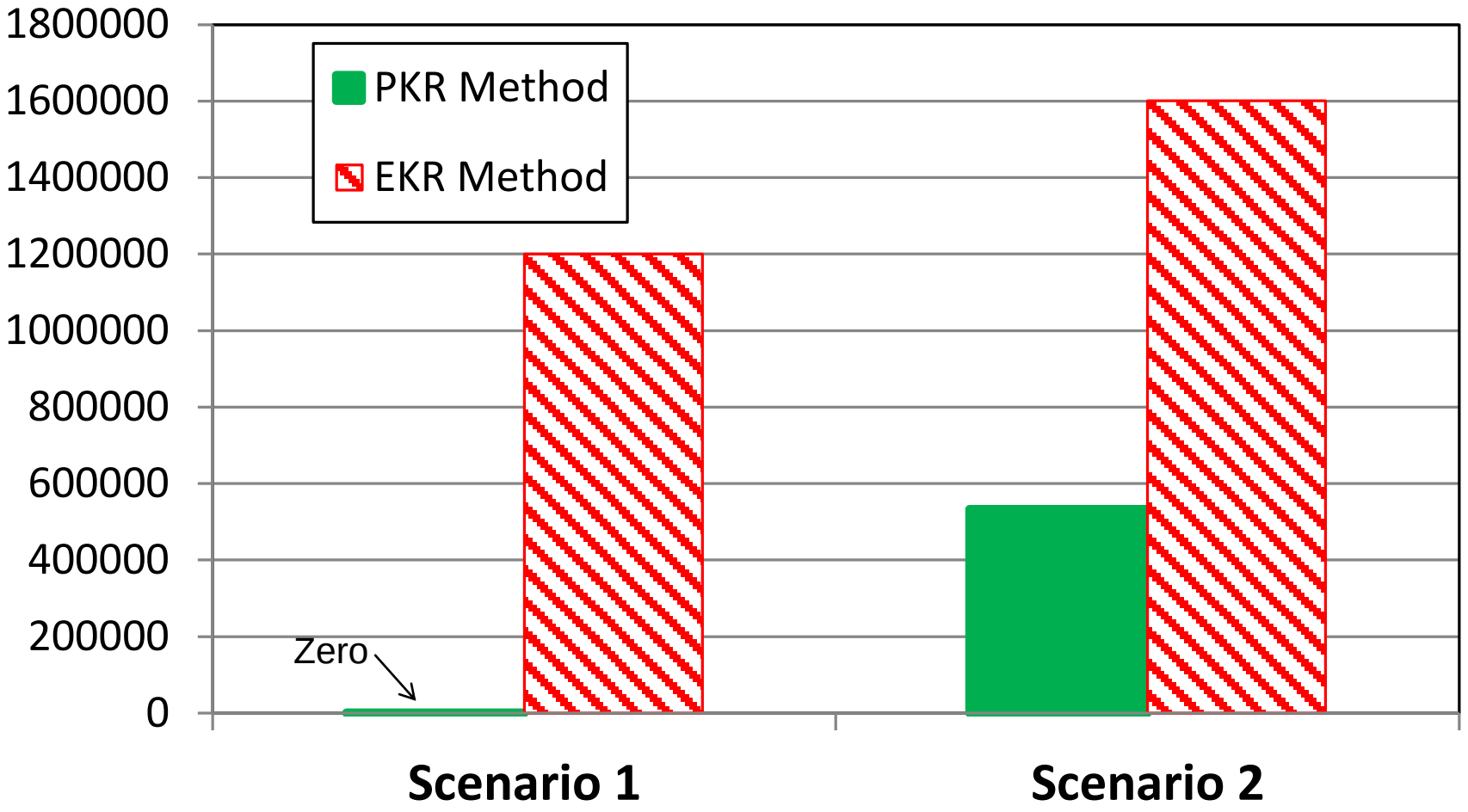
- Same assupmtions apply for AS1 through AS5 as in Scenario 1 except AS5 is multi-homed
- AS6 through AS8 give almost full table (400K signed prefixe updates) to AS5
- AS5 does not announce routes learned from one ISP to another (policy)
- Assume AS5's best path routes to the 400K prefixes are evenly distributed (i.e., 133.3K routes each) via AS6, AS7, and AS8
- Event: Peering between AS6 and AS5 is discontinued

Workload Comparison:

- When the peering (AS5-AS6) is discontinued:
  - ❖ In the PKR method, the router at AS5 signs and re-propagates 4x133.3K = 533K prefix updates in total
  - ❖ In contrast, in the EKR method (EKR-A or EKR-B), the router at AS5 signs and re-propagates 4x400K = 1.6 MILLION signed prefix updates

Summary of Comparison of PKR vs. EKR: Scenarios 1 & 2

Total # of Updates Signed and Re-propagated When Peering Change Event Occurs



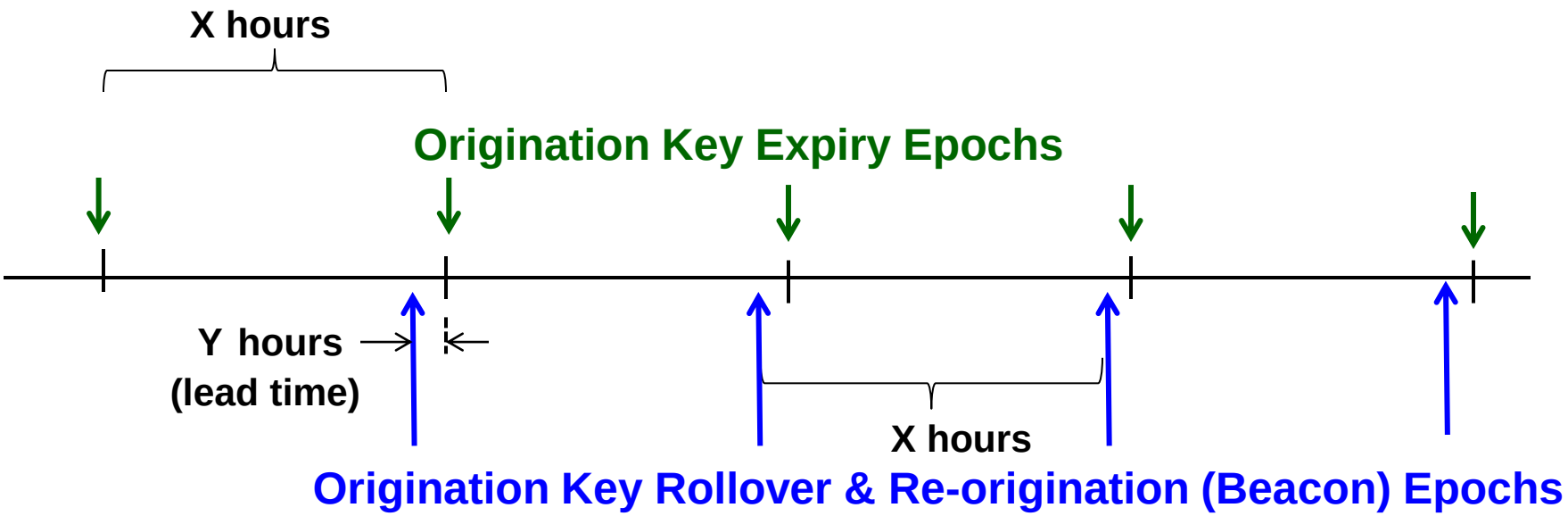
- BGPSEC with PKR generates the same number of prefix-route re-propagations as BGP-4 when a peering/policy change event occurs
- BGPSEC with EKR typically generates far more for the same scenario

Possible PKR & EKR Co-Existence

- The method described in draft-ietf-sidr-bgpsec-rollover-01 is PKR
- PKR is proactive on part of the prefix originators
  - Alleviates worry on part of the prefix originator about an AS in the middle of a path having a topology change
- EKR method is reactive on part of the AS that has a topology or policy change
  - AS in the middle could defensively do EKR (even if PKR is recommended for all ASes)
  - Alleviates worry on part of the AS operator that some of the transit prefixes' owners may not be participating in PKR
  - About 84% of all ASes are stub ASes; they may take time before becoming savvy w.r.t. key rollover
- Possibly PKR and EKR may co-exist as a result
- Hooks needed for EKR are a subset of those for PKR

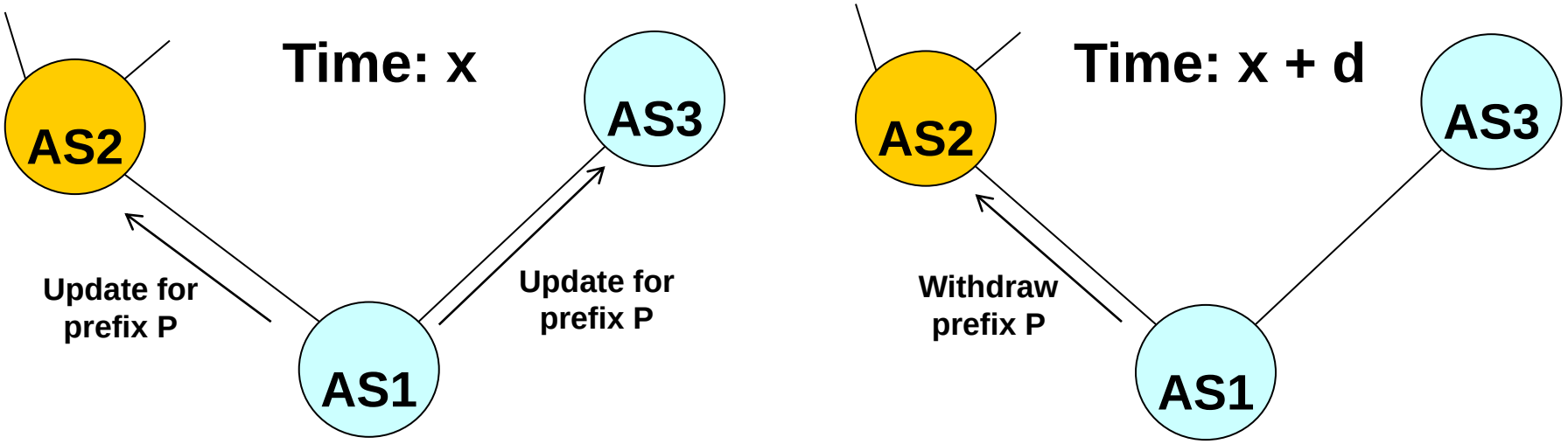
Backup slides

Time Flow Diagram to Explain PKR



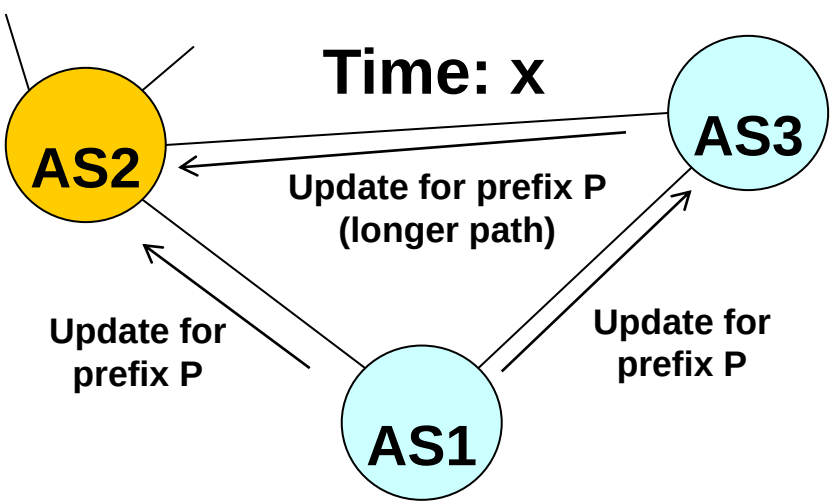
- A new pair of Current / Next keys comes into effect at each Origination Key Expiry Epoch
- Several sets of current/next origination keys can be propagated ahead of time to prepare the router in advance for several consecutive Origination Key Rollover & Re-origination (beacon) epochs

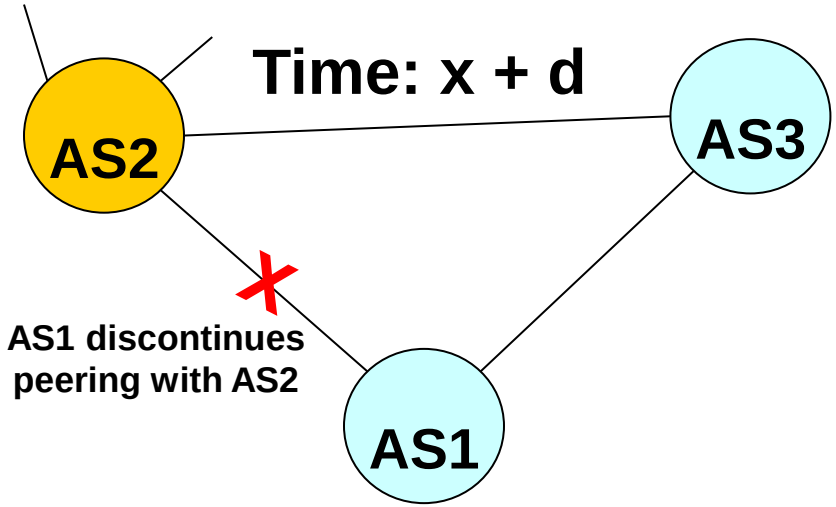
Replay Attack Example 1



- All AS peers here are eBGPSEC peers
- AS1 had announced a prefix P to AS2 at time x
- At a later time x+d, AS1 sends a Withdraw for prefix P to AS2
- AS2 suppresses the Withdraw (does not send to its peers any explicit or implicit Withdraw)

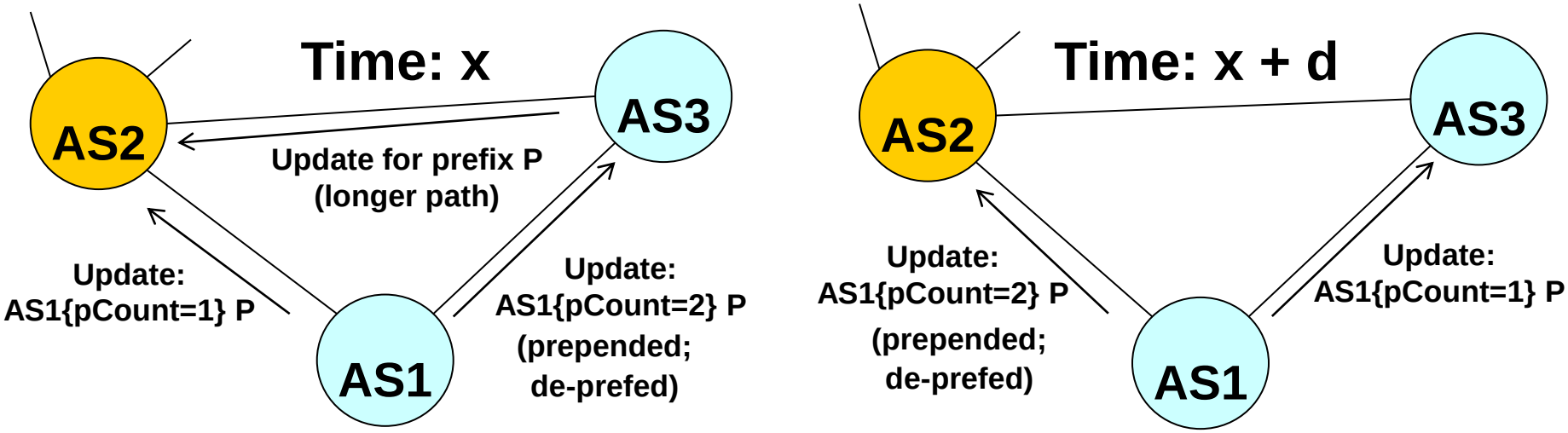
Replay Attack Example 2





- All AS peers here are eBGPSEC peers
- AS1 had announced a prefix P to AS2 at time x
- At a later time x+d, AS1 discontinues peering with AS2
- AS2 suppresses the Withdraw (does not send to its peers any explicit or implicit Withdraw)

Replay Attack Example 3



- All AS peers here are eBGPSEC peers
- AS1 had announced a prefix P; prefers ingress data path via AS2 over that via AS3
- At a later time x+d, AS1 switches ingress data path preference to AS3 over AS2
- AS2 suppresses the new prepended path announcement (does not send to its peers any update about P)