Secure Inter-Domain Routing
Internet-Draft
Intended status: Informational
Expires: April 21, 2016

K. Sriram
D. Montgomery
US NIST

October 19, 2015

Design Discussion and Comparison of Protection Mechanisms for Replay Attack and Withdrawal Suppression in BGPsec
draft-sriram-replay-protection-design-discussion-05

Abstract

In the context of BGPsec, a withdrawal suppression occurs when an adversary AS suppresses a prefix withdrawal with the intension of continuing to attract traffic for that prefix based on a previous (signed and valid) BGPsec announcement that was earlier propagated. Subsequently if the adversary AS had a BGPsec session reset with a neighboring BGPsec speaker and when the session is restored, the AS replays said previous BGPsec announcement (even though it was withdrawn), then such a replay action is called a replay attack. The BGPsec protocol should incorporate a method for protection from Replay Attack and Withdrawal Suppression (RAWS), at least to control the window of exposure. This informational document provides design discussion and comparison of multiple alternative RAWS protection mechanisms weighing their pros and cons. This is meant to be a companion document to the standards track I-D.-ietf-sidr-bgpsec-rollover that will specify a method to be used with BGPsec for RAWS protection.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 21, 2016.

Table of Contents

1. Introduction

In BGP or BGPsec, prefix or route withdrawals happen, and a withdrawal can be explicit (i.e. route simply withdrawn) or implicit (i.e. a new route announcement replaces the previous). In the context of BGPsec, a withdrawal suppression occurs when an adversary AS suppresses a prefix withdrawal with the intension of continuing to attract traffic for that prefix based on a previous (signed and valid) BGPsec announcement that was earlier propagated. Subsequently if the adversary AS has a BGPsec session reset with a neighboring

BGPsec speaker and when the session is restored, the AS replays said previous BGPsec announcement (even though it was withdrawn), then such a replay action is called a replay attack. The BGPsec protocol [I-D.ietf-sidr-bgpsec-protocol] requires a method for protection from Replay Attack and Withdrawal Suppression (RAWS), at least to control the window of exposure (see Sections 4.3, 4.4 of [RFC7353]).

In this informational document, we provide design discussion and comparison of various RAWS protection mechanisms that may be used in conjunction with the BGPsec protocol. This is meant to be a companion document to the standards track document [I-D.ietf-sidr-bgpsec-rollover] that will specify a method to be used with BGPsec for RAWS protection. Here we consider four alternative mechanisms - one based on the explicit Expiration Time approach and three variants based on the Key Rollover approach. We provide a detailed comparison among these mechanisms, weighing their pros and cons. This document is meant to help inform the decision process leading to an exact description for the mechanism to be finalized and formally specified in [I-D.ietf-sidr-bgpsec-rollover].

2. Description and Scenarios of Replay Attacks and Withdrawal Suppression

The following are examples of various forms of replay attack and withdrawal suppression (RAWS):

Example 1: AS1 has AS2 and AS3 as eBGPsec peers. At time x, AS1 had announced a prefix (P) to AS2 and AS3. 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). AS2 continues to attract some of the data for prefix P by pretending to still have a valid (signed) route for P. In effect, AS2 can conduct a Denial of Service (DOS) attack on a server located at prefix P. (See slide #15 in [RAWS-discussion] for an illustration.)

Example 2: AS1 has AS2 and AS3 as eBGPsec peers. AS2 and AS3 are also eBGPsec peers. At time x, AS1 announced a prefix P to AS2 and AS3. AS3 also propagates to AS2 its route (via AS1) for prefix P. At a later time (x+d), AS1 discontinues its peering with AS2. AS2 should propagate an alternate longer path via AS3 for prefix P and thus implicitly withdraw the route via AS1. However, AS2 suppresses it. AS2 can thus make some traffic destined for prefix P to flow via itself. This enables AS2 to eavesdrop on the data but not cause a DOS attack. AS2 may also choose to DoS attack hosts in prefix P. (See slide #16 in [RAWS-discussion] for an illustration.)

Example 3: AS1 has AS2 and AS3 as eBGPsec peers. AS2 and AS3 are also eBGPsec peers. At time x, AS1 announced a prefix P to AS2

without prepending (Update: AS1{pCount=1} P) but announced the same prefix to AS3 with prepending (Update: AS1{pCount=2} P). Thus AS1 had preferred its ingress data traffic for prefix P to come in via AS2. At a later time (x+d), AS1 switches ingress data path preference to AS3 over AS2 - announcing prefix P to AS3 without prepending (Update: AS1{pCount=1} P) and to AS2 with prepending (Update: AS1{pCount=2} P). AS2 suppresses the new prepended path announcement (does not send to its peers any new update about P). Thus AS2 continues to attract more of AS1's ingress data traffic and generates more revenue for itself at the expense of AS1. (See slide #17 in [RAWS-discussion] for an illustration.)

As illustrated above, the mechanisms and motivations for RAWS may differ.

In the context of the examples mentioned above, a requirement for RAWS protection can be stated as follows. An update that AS1 sends to AS2 at time x should expire at time x+w. This capability would allow other ASes to detect actions by AS2 to suppress the Withdraw or replay the update from AS1 for prefix P after time x+w. This limits the RAWS vulnerability window. (Note: If no peering or policy change affecting prefix P occurs during the vulnerability window, then a typical solution would include a method for extending the validity period of the route(s) beyond x+w.) We will later discuss what a reasonable window size, w, should be.

The obvious downside of any mechanism that support this capability is that it will require AS1 to send a new update before time x+w, and this update will need to propagate via all the paths that the original update traversed. Thus more update traffic will result than if the RAWS protection mechanism were not employed, and this traffic will require cryptographic processing by all of the routers along the paths. Thus the creation of a mechanism to counter RAWS attacks potentially introduces a new opportunity for DoS attacks against eBGPsec routers.

3. Classification of Solutions

Mechanisms for RAWS protection can be classified into two broad categories as follows:

- Expiration Time (ET) Method: This method uses an explicit Expiration Time field in the BGPsec update. (Note: Explicit Expire Time field was included in an earlier version of the BGPsec protocol specification [BGP draft-ietf-sidr-bgpsec-protocol-01].)

- Key Rollover (KR) Method: In this method, the update expiration is enforced by a key rollover. Router transitions to a new

certificate with a new pair of keys, and the previous router certificate either expires or is revoked.

The Key Rollover method can be further characterized into the following sub categories:

- Periodic Key Rollover (PKR): Key rollovers happen at periodic intervals.

- Event-driven Key Rollover (EKR): Key rollovers happen only when peering or policy change events occur.

  * EKR-A: EKR where expiry of previous update is enforced by CRL.

  * EKR-B: EKR where expiry of previous update is controlled by NotAfter time (router certificate is not revoked at the time when the event happens).

In Section 4, Section 5, and Section 6 we describe the various methods listed above, and discuss their pros and cons.

4. Expiration Time Method

The details of the Expiration Time (ET) method are as follow:

- Explicit Expiration Time is used for origin's signature.

- Expiration Time field is required in the BGPsec update.

- Periodic re-origination (beaconing) of prefixes is performed by origin ASes. The value in the ET field in the update is extended at beaconing time, and thereby the update is refreshed. Every prefix in the Internet is re-originated and propagates through the Internet once every 'beacon' interval.

- These beacons are distributed actions by prefix owners and are intended to be jittered in time to reduce burstiness. The beacon interval can be different at each originating AS.

- Beacon interval granularity: TBD but preferably in fairly granular units (days). It is important to limit the ability of each AS to specify a short beacon interval, to prevent an AS from using this mechanism to cause BGPsec to thrash.

Discussion of Pros and Cons:

Pro: This method is easy on transit routers. In the event of peering or policy change, BGPsec with the ET method behaves the same way as

BGP-4 in terms of which prefix routes are propagated. That is, the router re-evaluates best paths factoring in peering or policy changes, and propagates only those prefix routes that have a change in best path. In other words, there is no necessity for a transit BGPsec router to re-propagate and refresh prefixes on all peering links. This is because prefix updates are refreshed anyway once every beacon interval by all prefix originators. There is low steady-state traffic associated with beaconing (see Figure on slide #8 in [RAWS-discussion]), but there are no huge bursts or spikes in workload due to peering or policy change events at transit routers.

Con: Equipment vendor can potentially facilitate unnecessary frequent beaconing if ISP urges and pays (dollar attack!). This possibility is mitigated by having a well thought-out granularity for ET, for example, setting the unit for advertising ET to one day (rather than one minute).

Con: A change in on-the-wire BGPsec protocol would be needed in case the unit of the ET field (granularity) needs to be changed.

5. Key Rollover Method

Key Rollover (KR) method has three variations as outlined in Section 3. Those will be discussed later in this section. The following features are common to all variants of the KR method:

- In the KR method, it is best if the BGPsec router has two pairs of certificates as follows: A pair of origination certificates (current and next) for signing prefixes being originated by the AS of the router, and a pair of transit certificates (current and next) for signing 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 certificates and need not maintain the extra pair of transit certificates. (This would be the case for the vast majority of ASes, since most are stubs.)

- The three KR methods differ in how the rollover of certificates (or keys) is done:

  * Certificate rollovers are Periodic vs. Event-driven.

  * In the Event-driven method, the expiry of old update is (A) Enforced by CRL vs. (B) Controlled by NotAfter time.

  * In (A), certificate's NotAfter field is set to a very large value and CRL is issued to revoke the certificate when

necessary. In (B), NotAfter field set to a permissible vulnerability window time, and CRL to revoke certificate is not required.

Discussion of Pros and Cons (common to all Key Rollover methods):

Pro: The KR method functions by manipulating the RPKI objects (certificates, keys, NotAfter field in certificate, etc.) to refresh updates or to cause expiry of previously propagated updates. Unlike the ET method, it does not rely on any explicit field in the update. Hence, an advantage of the KR method over the ET method is that in case any parameters need to change or if the method itself is modified, then there is no impact on the BGPsec protocol on the wire.

Con: The KR method increases the number of objects in the RPKI repository system, by requiring at least two certificates for every transit AS. It also introduces additional churn in the global RPKI as these certificates expire (or are revoked) and are replaced.

Con: There is also added update churn. The amount of update churn varies depending on the type of KR method used (see Section 5.1 and Section 5.2).

We will now describe and discuss in detail the variants of the KR method.

5.1. Periodic Key Rollover Method

The details of the Periodic Key Rollover (PKR) method are as follow.

- Router's origination certificate's NotAfter time is used effectively as expiration time for origin's signature.

- Each origination router re-originates (i.e. beacons) before NotAfter time of the current origination certificate. Beaconing is periodic re-origination of prefixes by origin ASes.

- At beaconing time, the next origination certificate becomes the new current certificate, and the new update is signed with the private key of this new current certificate and re-originated.

- A new 'next' origination certificate is created and propagated at or before beaconing time. This can also be done with a good lead time. In practice, multiple 'next' certificates for each router could be propagated and kept in the in the RPKI repositories. They must have contiguous or slightly overlapping validity periods.

- Every prefix in the Internet is re-originated and propagates through the Internet once every 'beacon' interval.

- The re-originations or beacons are distributed actions by prefix owners and jittered in time by design to reduce burstiness. The beacon interval can be different at different originating ASes.

- Beacon (or re-origination) interval granularity: TBD but preferably in fairly granular units (days).

- Transit certificates can have large NotAfter time (e.g., whatever duration is required normally for key maintenance).

- When a peering or policy change event occurs at a transit router, the router does not perform any reactive key rollover. The router re-evaluates best paths factoring in peering or policy changes, and propagates only those prefix routes that have a change in best path (similar to BGP-4). There is no necessity for the BGPsec router to re-propagate and refresh prefixes on all peering links. This is because prefix updates are refreshed anyway once every re-origination (i.e. beaconing) interval by all prefix originators.

Discussion of Pros and Cons:

Several of the same pros/cons of the Expiration Time method also apply here for the PKR method.

Pro: The main pro for the PKR method is the same as that for the Expiration Time (ET) method. That is, being easy on transit routers as discussed in Section 4. Just as in the ET method, there is low steady-state traffic associated with periodic re-originations (i.e. beaconing) (see Figure on slide #8 in [RAWS-discussion]), but there are no huge bursts or spikes in workload due to peering or policy change events at transit routers. (See comparisons with the EKR methods in Section 5.2.)

Pro: The common pro discussed previously for all KR methods, namely, not requiring change of protocol on the wire when a parameter change occurs (e.g., change of beacon interval units) is naturally applicable here.

Con: Churn in the RPKI is of concern. Every BGPsec router renews and propagates its 'next' origination certificate once in every beacon (i.e. re-origination) interval.

5.2. Event-driven Key Rollover Method

The common details of the Event-driven Key Rollover (EKR) methods are as follow.

- Key rollover is reactive to events (not periodic).
- If a peering or policy change event involves only prefixes being originated at the AS of the router, then the router rolls only the origination key.
- If a peering change event involves transit prefixes at the AS of the router, then the router rolls its transit key as well as the origination key. Both keys are rolled because any peering relationship change also requires refresh of prefixes originated by the router.
- If a key rollover takes place, then a corresponding (origination or transit) new 'next' certificate is propagated in RPKI.

Discussion of Pros and Cons:

Pro: As long as no triggering events occur, there is no added update churn in BGPsec.

Con: Whenever the transit key is rolled, there is a storm of BGPsec updates at routers in transit ASes. For example, consider BGPsec capable transit AS5 that is connected to four BGPsec non-stub customers (AS1, AS2, AS3, AS4). Assume each AS has a single BGPsec router in it. AS1 through AS4 each receives almost full table (approximately 600K signed prefix updates) from AS5. Assume also that AS1 and its customers together originate 100 prefixes in total; likewise for AS2, AS3 and AS4. Now consider that an event occurs whereby the peering between AS1 and AS5 is discontinued. As a result of this event, in the EKR method, the AS5 router signs and re-propagates approximately 3x600K = 1.8 Million signed prefix updates to AS2, AS3 and AS4 combined. In addition, it also sends 4x100 = 400 Withdraws, which are negligible. In comparison, in the PKR method, reacting to the same event, the BGPsec router at AS5 sends only 4x100 = 400 Withdraws and signs/re-propagates ZERO prefix updates. (An illustration can be found in slide #9 in [RAWS-discussion]. Also, additional peering change scenarios and quantitative comparisons can be found in slides #10 and #11 in [RAWS-discussion].)

It remains to be seen through measurement and modeling how the impact of such large bursts of workload in the EKR method at the time of event occurrence can be managed in route processors, e.g., by jittering and throttling the workload.

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

EKR-A builds on the common principles as described for EKR above in Section 5.2. The additional details of EKR-A operation are as follow:

- NotAfter time of origination and transit certificates is set to a large value (e.g., one year or whatever period needed for normal key maintenance).

- Whenever key rollover (for origination or transit) occurs, then a CRL is propagated for the certificate that was used until that time. So the old update expires (due to invalid state) only when the CRL propagates and reaches each relying router.

- This method relies on end-to-end CRL propagation through the RPKI system to enforce expiry of a previous update whenever the need arises.

- The CRL either propagates all the way to the relying router, or the RPKI cache server of the router receives the CRL and then sends a withdrawal of the {AS, SKI, Pub Key} tuple to the router. Either way, the CRL must in effect propagate all the way to the relying router.

- Thus the attack vulnerability window with the EKR-A method is governed by the end-to-end CRL propagation time.

Discussion of Pros and Cons:

The following pro and con for the EKR-A method are in addition to the common pros and cons listed above for the KR and EKR methods (Section 5 and Section 5.2).

Pro: EKR-A has much less RPKI churn than PKR or EKR-B (see Section 5.2.2).

Con: Router needs to receive a CRL or a withdraw of {AS, SKI, Pub Key} tuple in order to know an update has expired. Hence, the RAWS vulnerability window is determined by the CRL propagation time which can vary widely from one relying router to another router that may be in different regions. It is anticipated that this would be no worse than 24 hours, but needs to be confirmed by measurements in an operational or emulated RPKI systems [rpki-delay].

5.2.2. EKR-B: EKR where Update Expiry is Enforced by NotAfter Time

EKR-B builds on the common principles as described for EKR above in Section 5.2. The additional details of EKR-B operation are as follow:

- NotAfter time of current origination and transit certificates is set to a value determined by the desired vulnerability window (~day).
- Update expiry is controlled by NotAfter time (router certificate is not revoked at the time when the event happens).
- If no triggering event occurs to cause origination key rollover within a pre-set time (NotAfter), then new origination (current and next) certificates are issued only to extend the NotAfter time but the corresponding key pairs and SKIs remain unchanged.
- Do likewise (i.e. similar to what the above bullet says) for the transit (current and next) certificates and keys.
- A previous update automatically becomes invalid at the earliest NotAfter time of the certificates used in the signatures unless each of those certificates' NotAfter time has been extended.
- Changes in certificates to extend their NotAfter time need not propagate end-to-end (all the way to the relying routers); they may propagate only up to the RPKI cache server of the relying router. RPKI cache server would send a withdraw for an {AS, SKI, Pub Key} tuple to a relying router if the NotAfter time of the certificate has passed.
- Changes in certificates to advance NotAfter time can be scheduled and propagated (in RPKI) reasonably well in advance.

Discussion of Pros and Cons:

The following pro and con for EKR-B are in addition to the common pros and cons listed above for the KR and EKR methods (Section 5 and Section 5.2).

Pro: Update expiration is automatic in case the NotAfter time of any of the certificates used to validate the update has not been extended. So the RAWS vulnerability window is predictable and not influenced by the RPKI end-to-end propagation time.

Pro: Routers do not get any RPKI updates from the RPKI cache server when a certificate changes but the corresponding key pair and SKI

remain unchanged. Routers do not receive NotAfter time from their RPKI cache server. There is no need for it. Instead, the RPKI cache server keeps track of NotAfter time, and provides to routers only valid {AS, SKI, Pub Key} tuples. This saves some RPKI state maintenance workload at the routers.

Con: EKR-B has much more RPKI churn than EKR-A because both origination and transit certificates need to be reissued periodically to extend their validity time (even in the absence of any peering or policy change events).

5.2.3. EKR with Separate Key for Each Incoming-Outgoing Peering-Pair

This is a place holder section where we mention another variant of the EKR method. This idea has not been considered or vetted by the SIDR WG yet. So we only mention it here briefly.

As noted earlier, the EKR methods considered so far generate a huge spike in workload whenever the transit key rollover takes place. One way to reduce that workload is to have a separate signing key for each incoming-outgoing peering pair. For example, consider a BGPsec router in AS4 that has peers in AS1, AS2, and AS3. The router will hold six signing keys, one each corresponding to (AS1, AS2), (AS2, AS1), (AS1, AS3), (AS3, AS1), (AS2, AS3), and (AS3, AS2) peering-pairs. Note that the directionality of peering is included here and is necessary. The key corresponding to (AS-i, AS-j) would only be used to sign updates received from AS-i and being forwarded to AS-j. In the general case, when the BGPsec router has n peers, the number of transit keys will be n(n-1). Since there would be a Current and a Next key (for rollover), the number of transit keys held in the router for signing will be actually 2n(n-1). When a peering or policy change occurs, the router would rollover only those specific keys that correspond to the peering-pairs over which the prefix updates are affected. In the above example, suppose a policy change between AS4 and AS1 causes AS4 to prepend prefixes sent to AS1 (pCount changed from 1 to 2). Then AS4 would do key rollover only for (AS2, AS1) and (AS3, AS1) peering-pairs, and not for any of the others. This would substantially reduce the quantity of prefix updates that are signed and re-propagated. In general, when peering or policy changes occur, this method will reduce the number of prefix updates to be re-propagated to exactly the same as that with normal BGP. That means that this method would also be on par with the ET and PKR methods in terms of update churn when a peering or policy change takes place. The downside of this method is that the router needs to maintain 2n(n-1) key pairs if it has n BGPsec peers.

Detailed discussion and comparison of this method with other methods can be provided in a later version of this document if the idea picks up interest in the WG.

6. Summary of Pros and Cons

Table 1 below summarizes the pros and cons for the various RAWS protection methods. This summary follows from the discussion above in Section 4 and Section 5.

Method	Pros	Cons
Expiration Time (ET)	1. The background load due to beaconing is low and not bursty. --- 2. Transit AS does NOT have a huge spike in workload even when a peering or policy change happens at that AS. Beaconing facilitates this. --- 3. Does not add to RPKI churn.	1. Prefix owner can abuse by beaconing too frequently. --- 2. Any change to the units (granularity) of ET field entails a change to on-the-wire BGPsec protocol. ---
Periodic Key Rollover (PKR)	1. The background load due to beaconing is low and not bursty. --- 2. Transit AS does NOT have a huge spike in workload even when a peering change happens at that AS. Beaconing (i.e. periodic re-origination) facilitates this. --- 3. If the periodic re-origination (i.e. beaconing) interval units change, BGPsec protocol	1. Prefix owner can abuse by beaconing (i.e. re-originating) too frequently. --- 2. Adds to RPKI churn. A pair of certificates (current and next) for each origination router are rolled once every beacon (i.e. re-origination) interval. Significantly more RPKI churn than that with EKR-A or EKR-B methods. ---

	on the wire remains unaffected. --- 4. Changes in the method (while still based on Key Rollover) can be accommodated without requiring any change to on-the-wire BGPsec protocol.	
---	---	---
Event driven Key Rollover Type A (EKR-A)	1. No update churn for long periods when no peering or policy changes occur. --- 2. The added churn in RPKI is much lower than that in the EKR-B method. --- 3. Same as Pro #4 for the PKR method.	1. Whenever the transit key is rolled (in response to a peering or policy change event), there is a storm of BGPsec updates, especially at routers in large transit ASes. --- 2. The RAWS vulnerability window is dependent on end-to-end CRL propagation. It may vary significantly from one relying router to another that may be in different regions. ---
---	---	---
Event driven Key Rollover Type B (EKR-B)	1. Same as Pro #1 for the EKR-A method. --- 2. The RAWS vulnerability window is enforced by NotAfter time in certificates and is therefore predictable. --- 3. Same as Pro #4 for the PKR method.	1. Same as Con #1 for the EKR-A method. --- 2. The added churn in RPKI is much higher than that in the EKR-A method. ---

Table 1: Table with Summary of Pros and Cons

7. Summary and Conclusions

We have attempted to provide insights into the operation of multiple alternative methods for RAWS protection. It is hoped that the SIDR WG will utilize the analysis presented here as input for deciding on the choice of a mechanism for protection from RAWS. Once that decision is made, the chosen mechanism would be included in the standards track document [I-D.ietf-sidr-bgpsec-rollover].

Some important considerations for the decision making can be possibly listed as follow:

1. The Expiration Time (ET) method is best (on par with the PKR method) in terms of preventing huge update workloads during peering and policy change events at transit routers with several peers. It has no added RPKI churn. But the ET method has the disadvantage of requiring on-the-wire protocol change if some parameters (e.g., the units of beacon interval) change.

2. The Periodic Key Rollover (PKR) method operates the same way as the ET method for preventing huge update workloads during peering and policy change events at transit routers with several peers. It does not have the disadvantage of requiring on-the-wire protocol change if some parameters (e.g., the units of beaconing/ re-origination periodicity) change. But it has the downside of added RPKI churn.

3. The Event-driven Key Roll (EKR-A and EKR-B) methods have significantly less RPKI churn than the PKR method. They also have no BGPsec update churn during long quiet periods when no peering or policy change events occur. But they suffer the drawback of creating huge update workloads during peering and policy change events at transit routers with several peers. Can this workload be jittered or flow controlled to spread it over time without convergence delay concerns? May be - needs further study.

4. The EKR-A method relies on end-to-end CRL propagation through the RPKI system to enforce expiry of a previous update when needed. By contrast, in the EKR-B method the update expiry is controlled by NotAfter time of the certificates used in update signatures. In EKR-B method, previous update automatically becomes invalid at the earliest NotAfter time of the certificates used in the signatures unless each of those certificates' NotAfter time has been extended. Also, in EKR-B method, changes in certificates to extend their NotAfter time need not propagate end-to-end (all the

way to the relying routers); they may propagate only up to the RPKI cache server of the relying router (see Section 5.2.2). The changes in certificates to advance NotAfter time can be scheduled and propagated (in RPKI) reasonably well in advance.

5. Besides being out-of-band relative to the BGPsec protocol on the wire, the other good thing about the Key Rollover method is that once the basics of the mechanism are implemented, there may be flexibility to implement PKR, EKR-A or EKR-B on top of it. It may also be possible to switch from one method to another (within this class) if necessary based on operational experience; this transition would not require any change to on-the-wire BGPsec protocol.

8. Acknowledgements

The authors would like to thank Steve Kent for extensive review and many useful suggestions on an earlier version of this document. Thanks are also due to Roque Gagliano and Brian Weis for helpful discussions. Further, we are thankful to Oliver Borchert and Okhee Kim for comments and suggestions.

9. IANA Considerations

This memo includes no request to IANA.

10. Security Considerations

This memo requires no security considerations of its own since it is targeted to be an informational RFC in support of [I-D.ietf-sidr-bgpsec-rollover] and [I-D.ietf-sidr-bgpsec-protocol] . The reader is therefore directed to the security considerations provided in those documents.

11. Informative References

[I-D.ietf-sidr-bgpsec-protocol]
Lepinski, M., "BGPsec Protocol Specification", draft-ietf-sidr-bgpsec-protocol-13 (work in progress), July 2015.

[I-D.ietf-sidr-bgpsec-rollover]
Gagliano, R., Patel, K., and B. Weis, "BGPsec Router Certificate Rollover", draft-ietf-sidr-bgpsec-rollover-04 (work in progress), July 2015.

[RAWS-discussion]
Sriram, K. and D. Montgomery, "Discussion of Key Rollover Mechanisms for Replay-Attack Protection", Presented at IETF-85 SIDR WG Meeting, November 2012, <http://www.ietf.org/proceedings/85/slides/slides-85-sidr-4.pdf>.

[RFC7353] Bellovin, S., Bush, R., and D. Ward, "Security Requirements for BGP Path Validation", RFC 7353, DOI 10.17487/RFC7353, August 2014, <http://www.rfc-editor.org/info/rfc7353>.

[rpki-delay]
Kent, S. and K. Sriram, "RPKI rsync Download Delay Modeling", Presented at IETF-86 SIDR WG Meeting, March 2013, <http://www.ietf.org/proceedings/86/slides/slides-86-sidr-1.pdf>.

Authors' Addresses

Kotikalapudi Sriram
US NIST

Email: ksriram@nist.gov

Doug Montgomery
US NIST

Email: dougm@nist.gov