Flexible and Robust Key Rollover in DNSSEC

Yuri Schaeffer, Matthijs Mekking, and Benno Overeinder

NLnet Labs

A bouillabaisse or some other stew?

WHAT IS FOR DINNER?

Ingredients of This Presentation

- DNS: My data is controlled by me. I can change it to whatever I want *whenever* I want.
- TTL: I *promise* this data is useable for the next *N* seconds.
- DNSSEC: Replace a bit of freedom with security.

Degraded Trust

- I lost trust in key X,
  let me switch it for key Y.
- Keys need to be rolled, in such a way that chain of trust is not broken.
- Recipes for rolling are documented.

Example Roll

Roll from ZSK A to ZSK B:

1. Publish $DNSKEY_B$
2. Wait $TTL(DNSKEY)$
3. Switch $RRSIG_A$ for $RRSIG_B$
4. Wait $TTL(RRSIG)$
5. Remove $DNSKEY_A$
6. Wait $TTL(DNSKEY)$

THE RECIPE

The Recipe

- Question

  Give me the recipe for rolling a key.

- Answer

  KSK, ZSK, CSK.

Another Recipe

- Question
  I need to change algorithm as well.
- Answer
  KSK, ZSK, CSK, Algorithm KSK, Algorithm ZSK, Algorithm CSK.

And Another Recipe

- Question
  How about switching between split key and single key?
- Answer
  KSK, ZSK, CSK, Algorithm KSK, Algorithm ZSK, Algorithm CSK, Split to single, Single to split, Algorithm Split to single, Algorithm single to split.

And Another …

- Question
  I'd like to minimize parent interactions.
- Answer
  KSK, ZSK, CSK, Algorithm KSK, Algorithm ZSK, Algorithm CSK, Split to single, Single to split, Algorithm Split to single, Algorithm single to split, Minimize DS KSK, Minimize DS CSK, Minimize DS Split to single, Minimize DS single to split, Minimize DS Algorithm KSK, Minimize DS Algorithm CSK, Minimize DS Algorithm Split to single, Minimize DS Algorithm single to split.

And Another …

- Question
  Or minimize the DNSKEY set.
- Answer
  KSK, ZSK, CSK, Algorithm KSK, Algorithm ZSK, Algorithm CSK, Split to single, Single to split, Algorithm Split to single, Algorithm single to split, Minimize DS KSK, Minimize DS CSK, Minimize DS Split to single, Minimize DS single to split, Minimize DS Algorithm KSK, Minimize DS Algorithm CSK, Minimize DS Algorithm Split to single, Minimize DS Algorithm single to split, Minimize DNSKEY KSK, Minimize DNSKEY ZSK, Minimize DNSKEY CSK, Minimize DNSKEY Split to single, Minimize DNSKEY single to split, Minimize DNSKEY Algorithm KSK, Minimize DNSKEY Algorithm ZSK, Minimize DNSKEY Algorithm CSK, Minimize DNSKEY Algorithm Split to

And Another …

- Question
  Or minimize the signatures.
- Answer
  KSK, ZSK, CSK, Algorithm KSK, Algorithm ZSK, Algorithm CSK, Split to single, Single to split, Algorithm Split to single, Algorithm single to split, Minimize DS KSK, Minimize DS CSK, Minimize DS Split to single, Minimize DS single to split, Minimize DS Algorithm KSK, Minimize DS Algorithm CSK, Minimize DS Algorithm Split to single, Minimize DS Algorithm single to split, Minimize DNSKEY KSK, Minimize DNSKEY ZSK, Minimize DNSKEY CSK, Minimize DNSKEY Split to single, Minimize DNSKEY single to split, Minimize DNSKEY Algorithm KSK, Minimize DNSKEY Algorithm ZSK, Minimize DNSKEY Algorithm CSK, Minimize DNSKEY Algorithm Split to

And Another …

- Question
  Or two of those at the same time!
- Answer
  KSK, ZSK, CSK, Algorithm KSK, Algorithm ZSK, Algorithm CSK, Split to single, Single to split, Algorithm Split to single, Algorithm single to split, Minimize DS KSK, Minimize DS CSK, Minimize DS Split to single, Minimize DS single to split, Minimize DS Algorithm KSK, Minimize DS Algorithm CSK, Minimize DS Algorithm Split to single, Minimize DS Algorithm single to split, Minimize DNSKEY KSK, Minimize DNSKEY ZSK, Minimize DNSKEY CSK, Minimize DNSKEY Split to single, Minimize DNSKEY single to split, Minimize DNSKEY Algorithm KSK, Minimize DNSKEY Algorithm ZSK, Minimize DNSKEY Algorithm CSK, Minimize DNSKEY Algorithm Split to

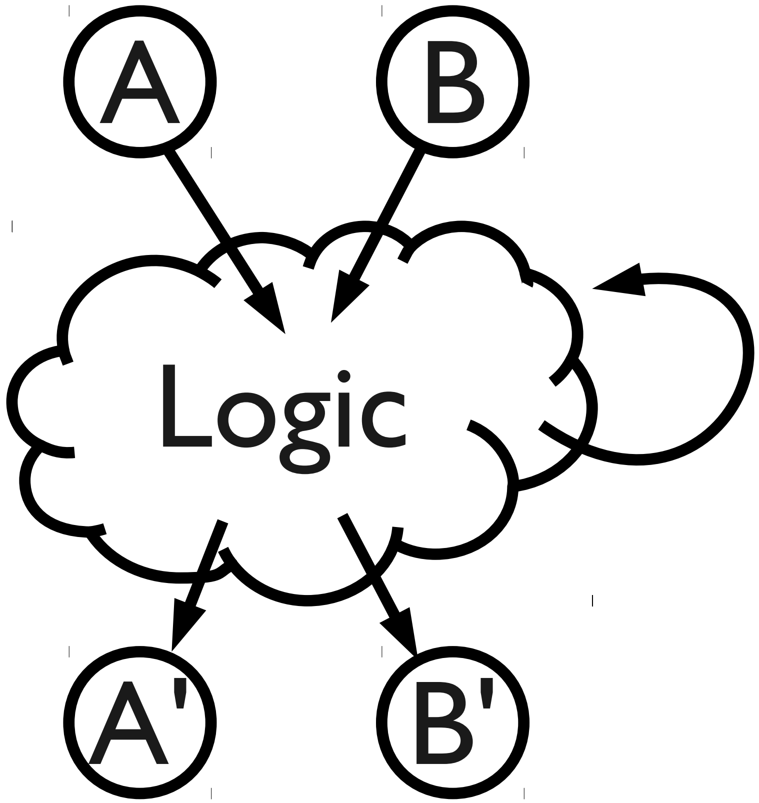
And Another …

- Question
  What can we do in case of an emergency?
- Answer
  KSK, ZSK, CSK, Algorithm KSK, Algorithm ZSK, Algorithm CSK, Split to single, Single to split, Algorithm Split to single, Algorithm single to split, Minimize DS KSK, Minimize DS CSK, Minimize DS Split to single, Minimize DS single to split, Minimize DS Algorithm KSK, Minimize DS Algorithm CSK, Minimize DS Algorithm Split to single, Minimize DS Algorithm single to split, Minimize DNSKEY KSK, Minimize DNSKEY ZSK, Minimize DNSKEY CSK, Minimize DNSKEY Split to single, Minimize DNSKEY single to split, Minimize DNSKEY Algorithm KSK, Minimize DNSKEY Algorithm ZSK, Minimize DNSKEY Algorithm CSK, Minimize DNSKEY Algorithm Split to

…

Right. I'll just implement these two.

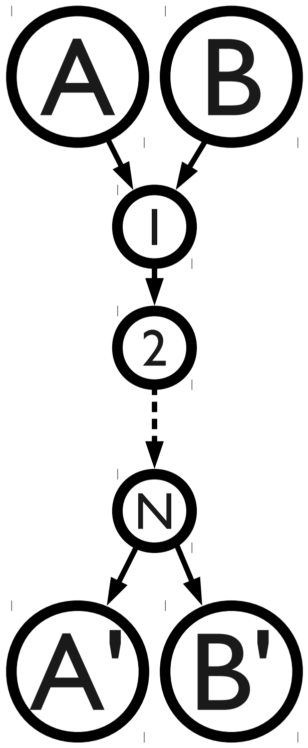
KSK, ZSK.



COOKING 101

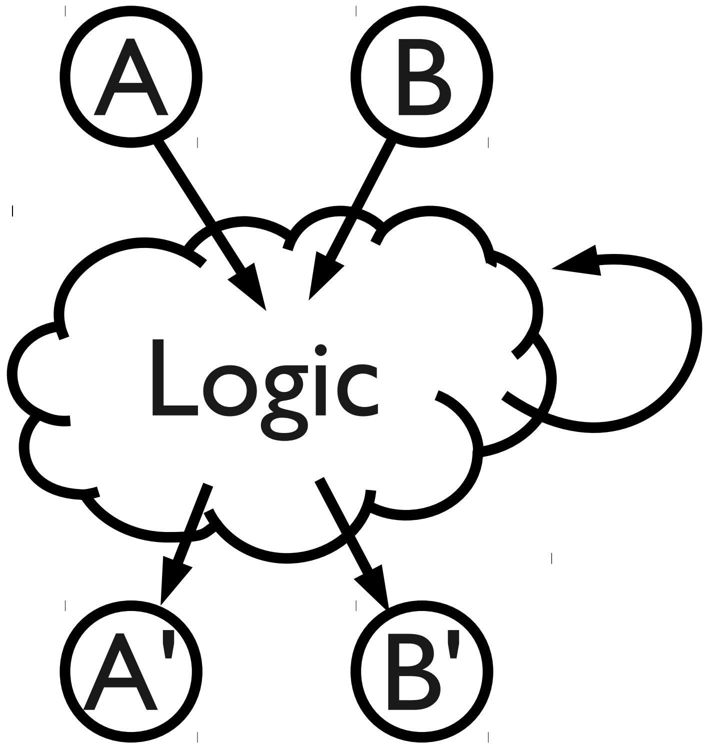
NLnet Labs

ZSK Prepublication Rollover

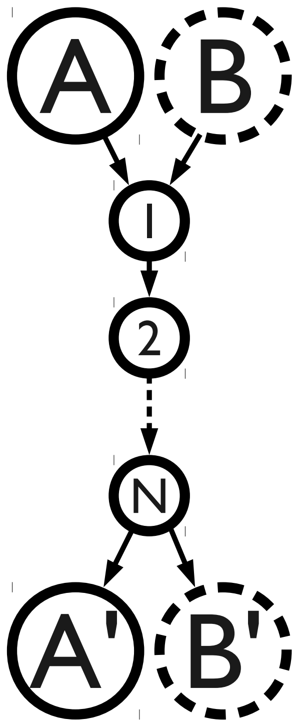

$$\begin{tikzcd}
A \arrow[dr] & & B \arrow[dl] \\
& 1 \arrow[d] & \\
& 2 \arrow[d, dashed] & \\
& N \arrow[dl] \arrow[dr] & \\
A' & & B'
\end{tikzcd}$$

Recipe:

1. Publish $DNSKEY_B$
2. Wait $TTL(DNSKEY)$
3. Switch $RRSIG_A$ for $RRSIG_B$
4. Wait $TTL(RRSIG)$
5. Remove $DNSKEY_A$
6. Wait $TTL(DNSKEY)$

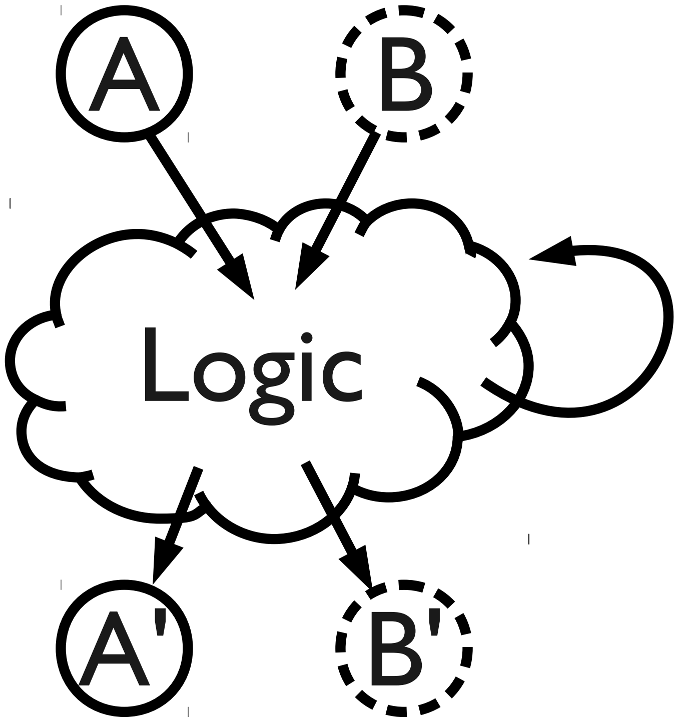

$$\begin{tikzcd}
A \arrow[dr] & & B \arrow[dl] \\
& \text{Logic} \arrow[dl] \arrow[dr] \arrow[loop right] & \\
A' & & B'
\end{tikzcd}$$

CSK Algorithm Rollover



Recipe:

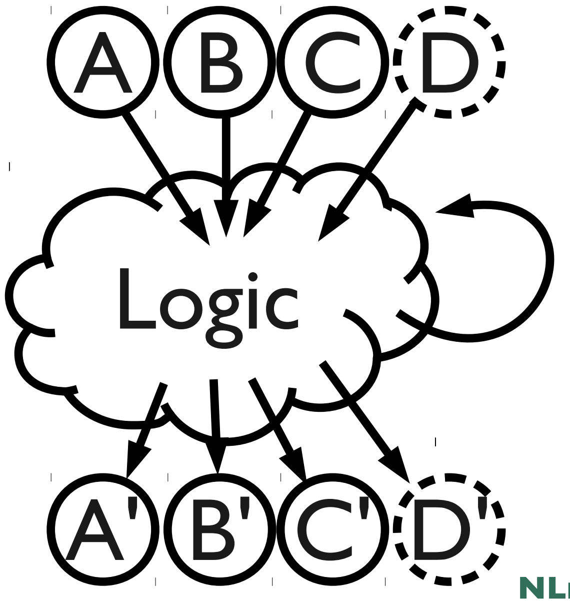
1. Publish $RRSIG_B$
2. Wait $TTL(RRSIG)$
3. Publish $DNSKEY_B$
4. Wait $TTL(DNSKEY)$
5. Switch DS_A for DS_B
6. Wait $TTL(DS)$
7. Remove $DNSKEY_A$
8. Wait $TTL(DNSKEY)$
9. Remove $RRSIG_A$
10. Wait $TTL(RRSIG)$



KSK+ZSK to CSK of different algorithm while minimizing published DS and RRSIG records and breaking up an earlier rollover -rollover.

?

…



A dataflow-oriented approach

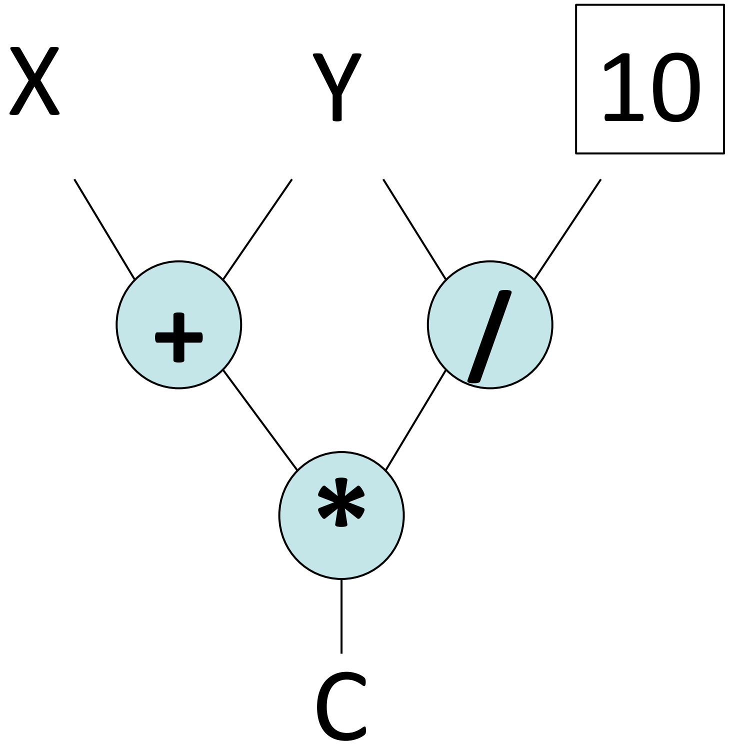
ROLLOVER CENTRISM VS. KEY CENTRISM

The Dataflow Model

A := X + Y

B := Y / 10

C := A * B

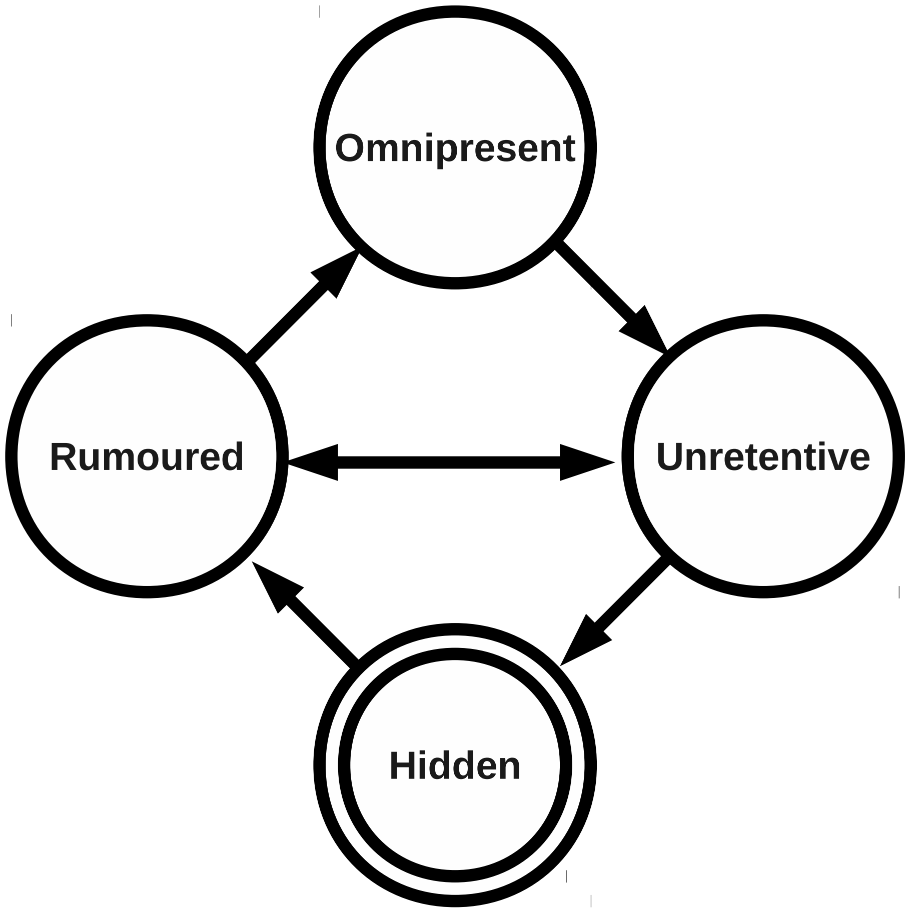


- Dataflow as a coordination language, like Linda [Gelernter and Carriero 1992]

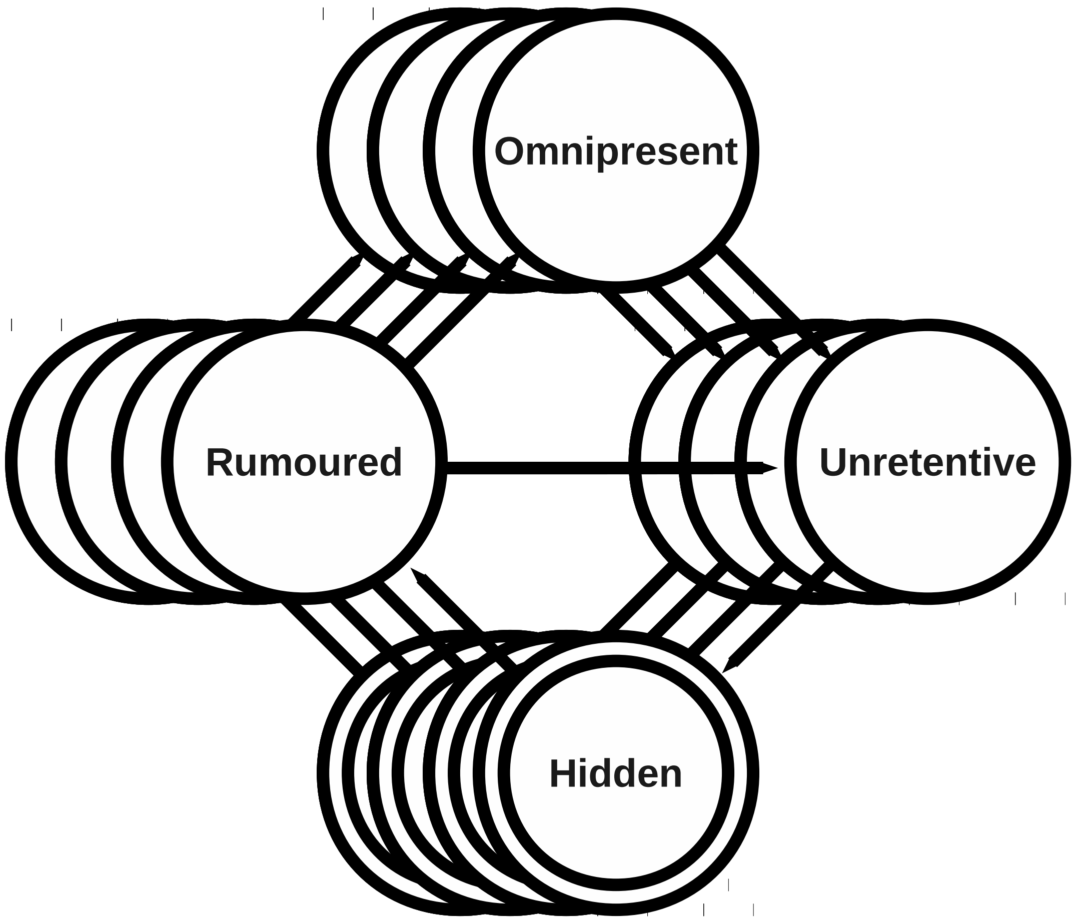
The Key Rollover Stew Recipe

- Define a set of valid key states
- Specify conditions whether a transition results in a valid key state
- Give goals that have to be realized in the rollover of a key
- Let the machinery find the shortest safe path towards a desired state.

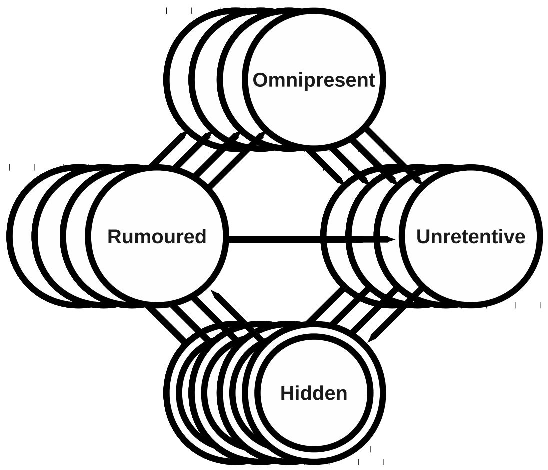
The Key Rollover State Machine

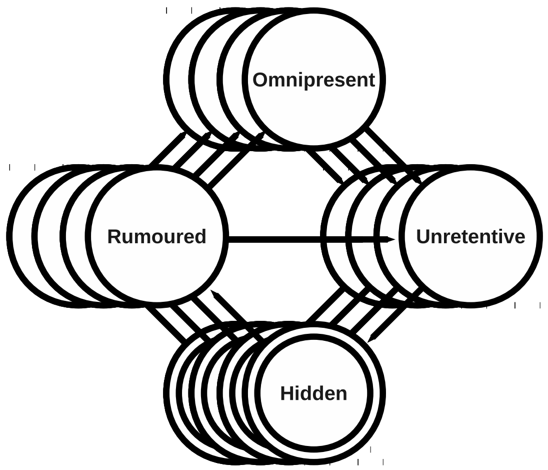


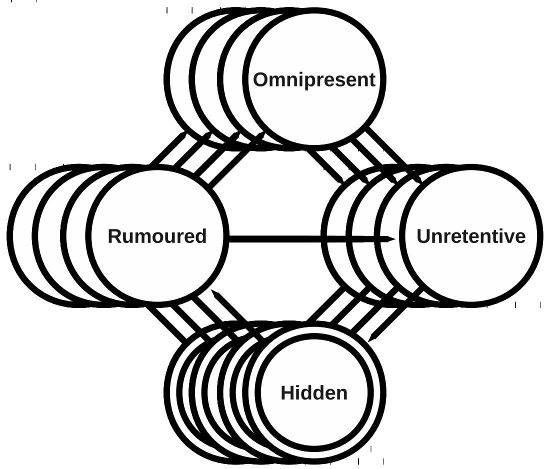
DS, DNSKEY, RRSIG DNSKEY, and RRSIG RR Seperated



Generalized Model for DNSSEC Key Rollover







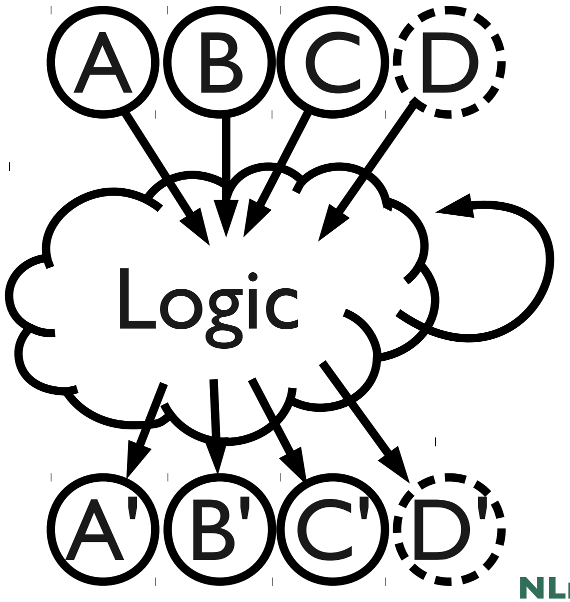
Formalize Validity of a Zone

$$
\begin{array}{llr}
\mathit{rule}1(x): & & \\
\quad \exists y \in K\,(D_y^{\uparrow +}) & & (1a) \\
 & & \\
\mathit{rule}2(x): & & \\
\quad \exists y \in X\,(D_y^{+} K_y^{+} R_y^{+}) & \vee & (1b) \\
\quad \exists y, z \in X\,(D_y^{\uparrow} K_y^{+} R_y^{+} D_z^{\downarrow} K_z^{+} R_z^{+} \wedge y \overset{D}{\succ} z) & \vee & (1c) \\
\quad \exists y, z \in X\,(D_y^{+} K_y^{\uparrow +} R_y^{\uparrow} D_z^{+} K_z^{\downarrow} R_z^{\downarrow -} \wedge y \overset{K}{\succ} z) & \vee & (1d) \\
\quad \forall y \in X\,(D_y^{-} \vee \exists z \in X\,(K_z^{+} R_z^{+} (D_y = D_z))) & & (1e) \\
 & & \\
\mathit{rule}3(x): & & \\
\quad \exists y \in X\,(K_y^{+} S_y^{+}) & \vee & (1f) \\
\quad \exists y, z \in X\,(K_y^{\uparrow} S_y^{+} K_z^{\downarrow} S_z^{+} \wedge y \overset{K}{\succ} z) & \vee & (1g) \\
\quad \exists y, z \in X\,(K_y^{+} S_y^{\uparrow} K_z^{+} S_z^{\downarrow} \wedge y \overset{S}{\succ} z) & \vee & (1h) \\
\quad \forall y \in X\,(K_y^{-} \vee \exists z \in X\,(S_z^{+} (K_y = K_z))) & & (1i)
\end{array}
$$

KSK+ZSK to CSK of different algorithm while minimizing published DS and RRSIG records and breaking up an earlier rollover -rollover.

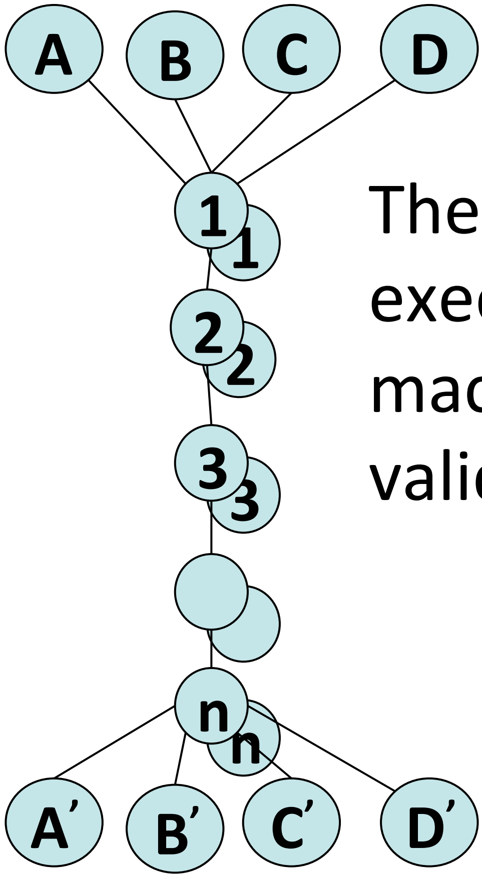
?

...

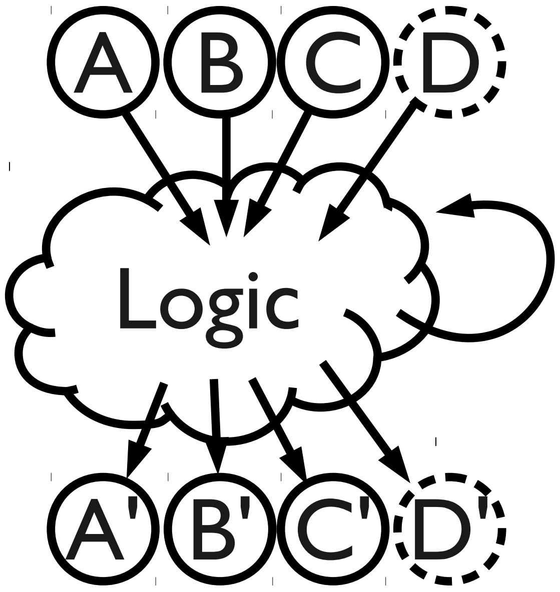


NLnet Labs

Dataflow-Driven State Machine Model



The dataflow-driven execution of the state machines generates a valid key rollover path



NLnet Labs

Concluding Remarks

- Prototype generates valid key rollover paths
- The key centric rollover mechanism will be part of OpenDNSSEC (Enforcer in OpenDNSSEC 2.0)
- ZKS, KSK, CSK, algorithm rollover supported
- Ability to switch between ongoing rollovers: emergency key rollover