



## DNSSEC CENSUS: ARE DNSKEY TRANSITIONS WORKING?

Eric Osterweil - [eoster@gmu.edu](mailto:eoster@gmu.edu) - Assistant Professor, Computer Science

Steve Crocker - [steve@shinkuro.com](mailto:steve@shinkuro.com) - Shinkuro

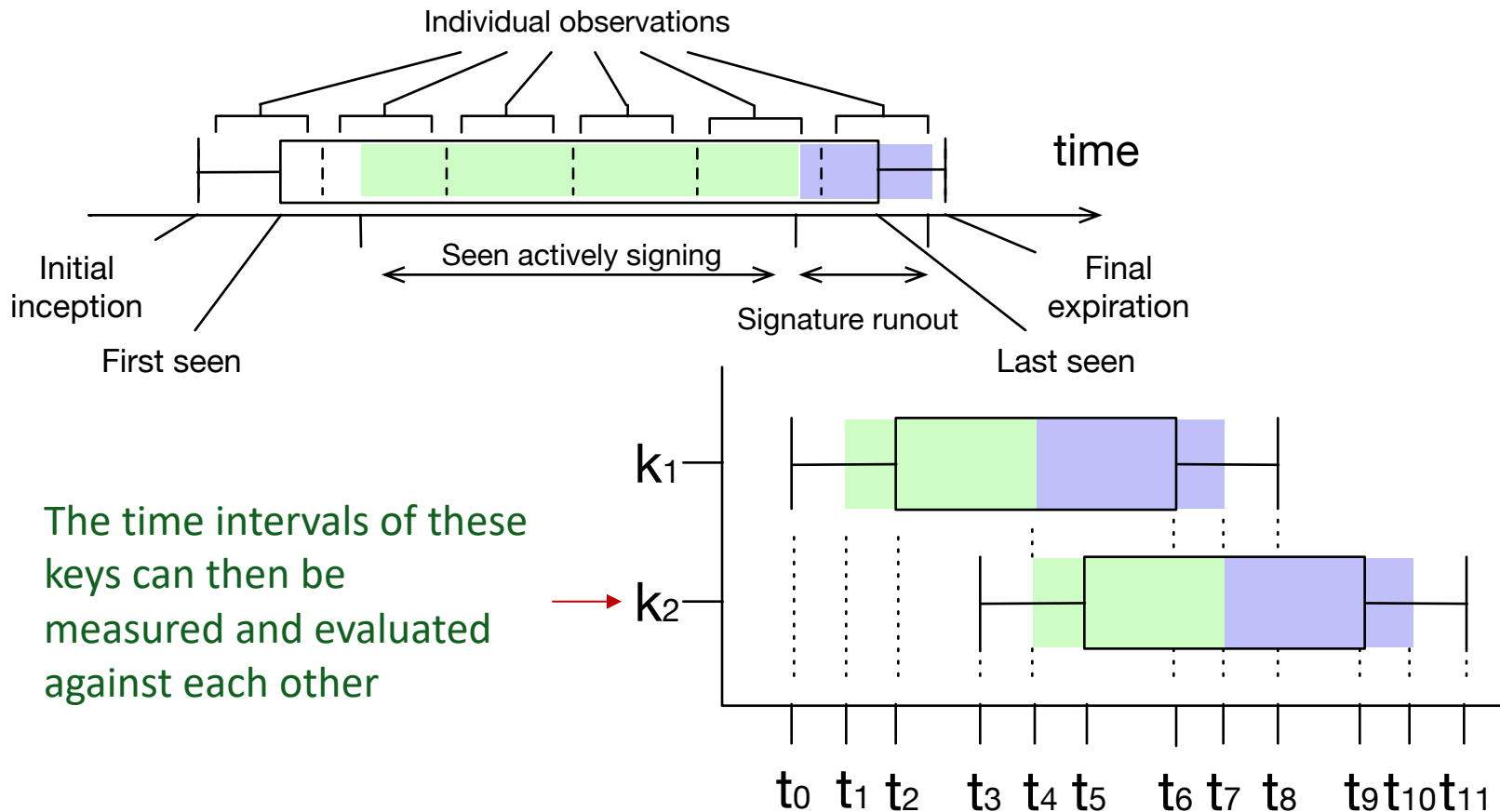


## INTRODUCTION

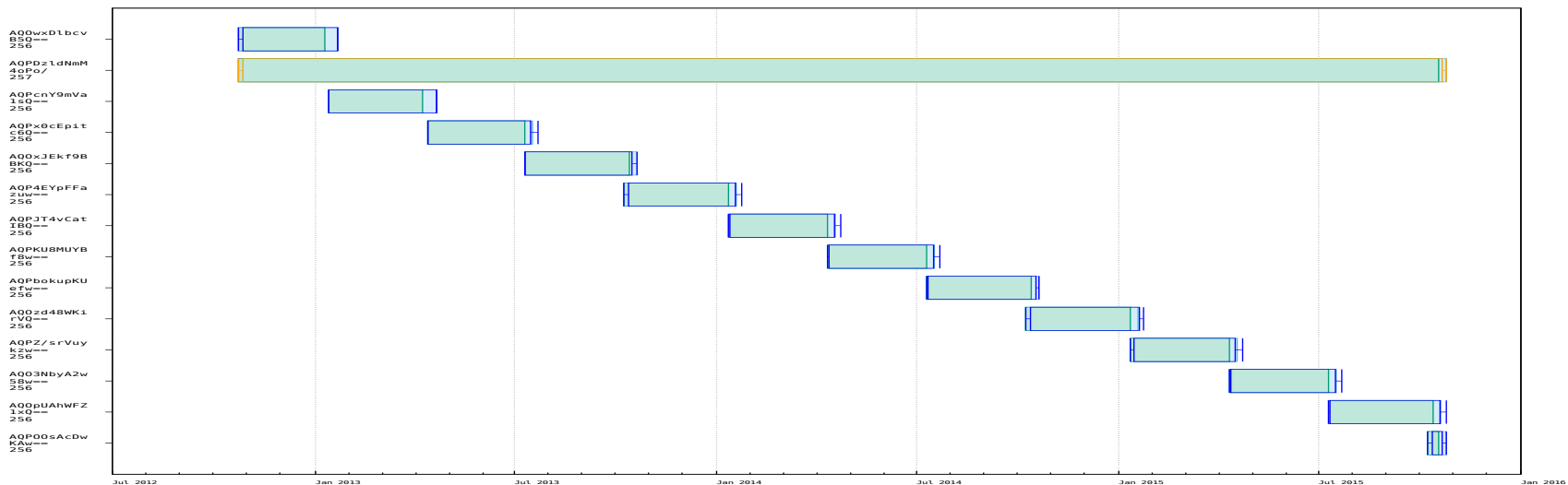
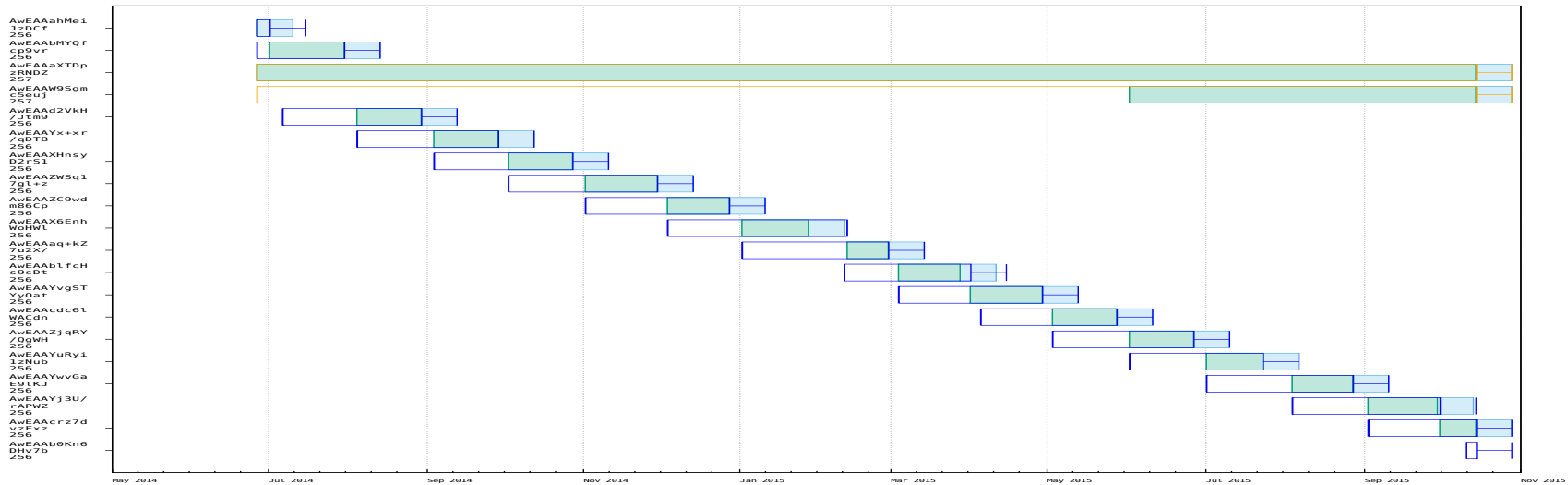
- DNSSEC's protections stem from DNSKEYs (and DSes), and guidance dictates that we periodically change them
  - Guidance on “key rollovers” has been evolving for many years: RFC-5011, RFC-7583, etc.
  - The Root zone's KSK was just rolled!
- But, what do key rollovers *look* like, and are they “working?”
- In this talk, I will present a framework for evaluating just that: “what *are* a key rollovers, and can we evaluate them?”
- But first, some pedanticism... Are “rollovers” when a single key changes to another key (a *1:1* transition)
- Well then, if a zone transitions from *n* keys to *m* keys, which key(s) rolled over to which other keys?
  - Did all disappearing keys rollover to each/all of the remaining keys?
  - If only some other keys remained get used, did they get rolled over to as well?
- We propose “*key transition*” is the general superset of key rollovers
  - That is, a degenerative case of an *n:m* transition may be a *1:1* rollover

# BRIDGING, BUSTING, AND BINDING METHODOLOGY: FROM MEASUREMENTS TO A MODEL

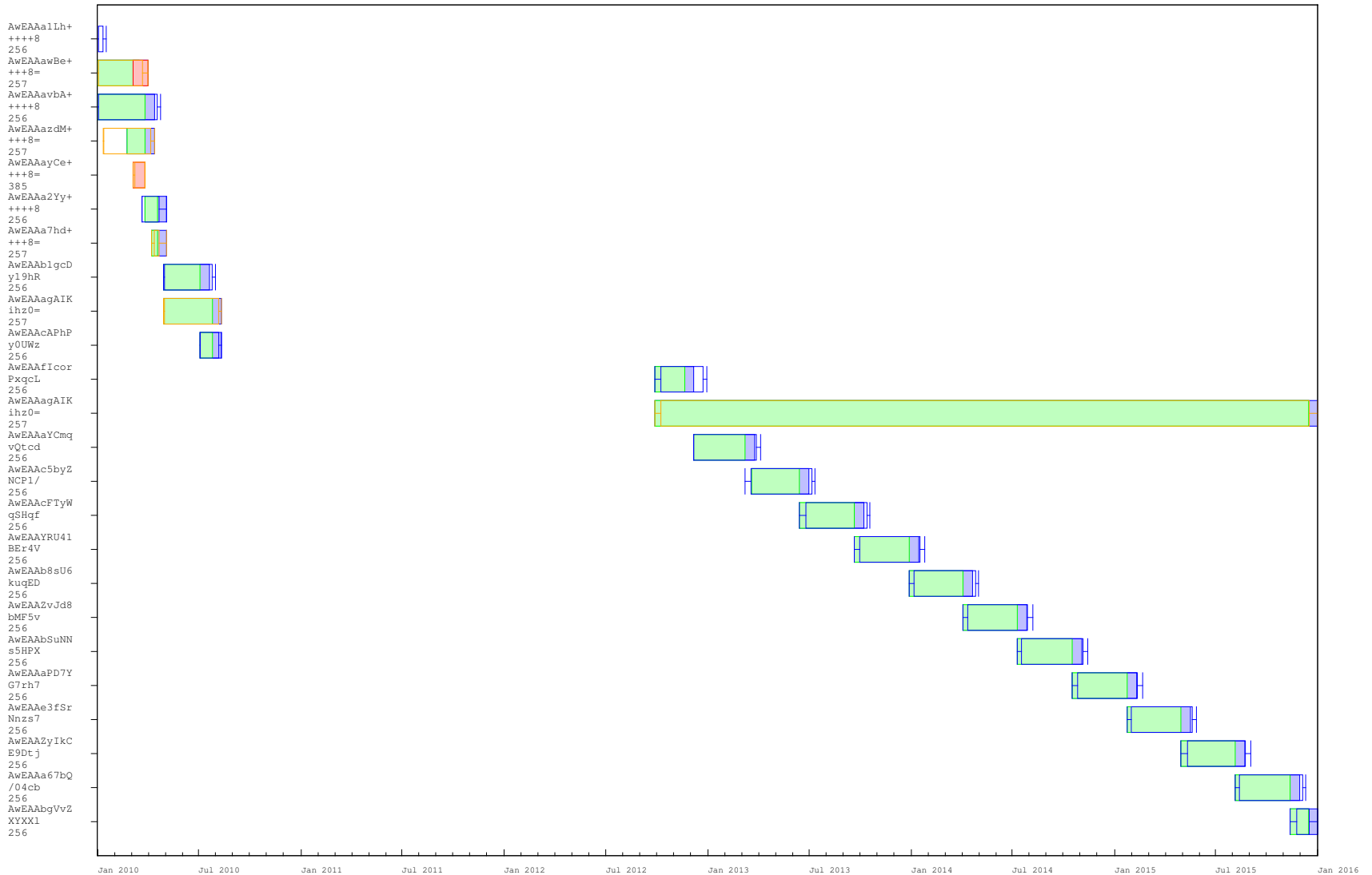
- As photo snapshots can be projected into video, measurements must become models
- Bridged and Busted observations are the **Bound** into longitudinal key entities



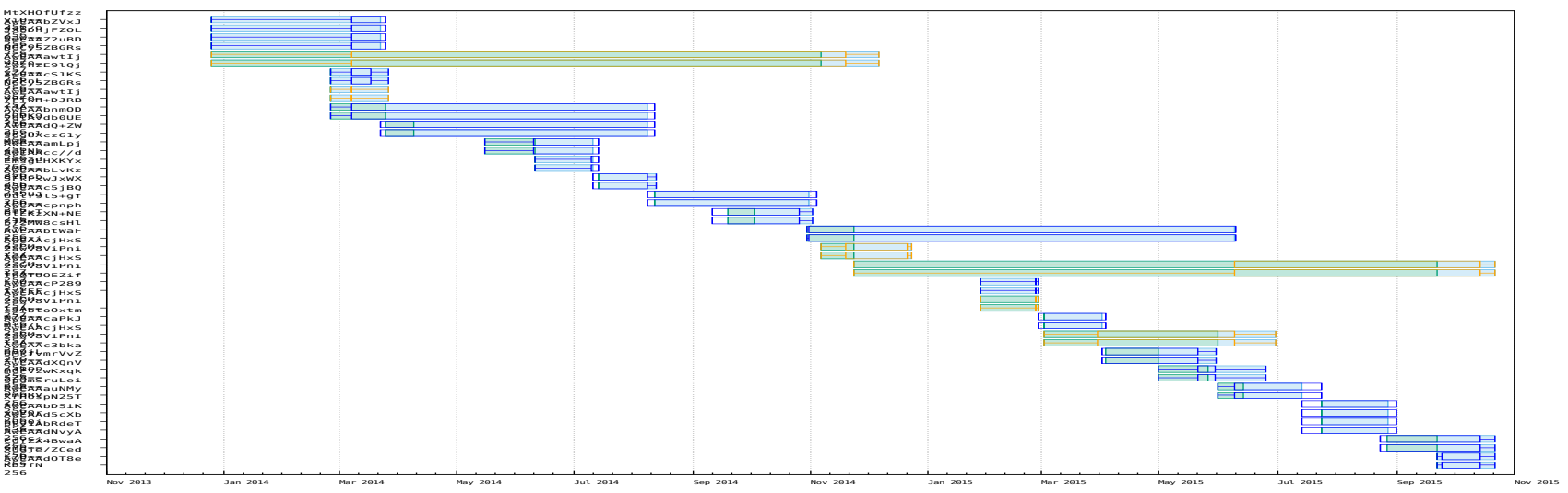
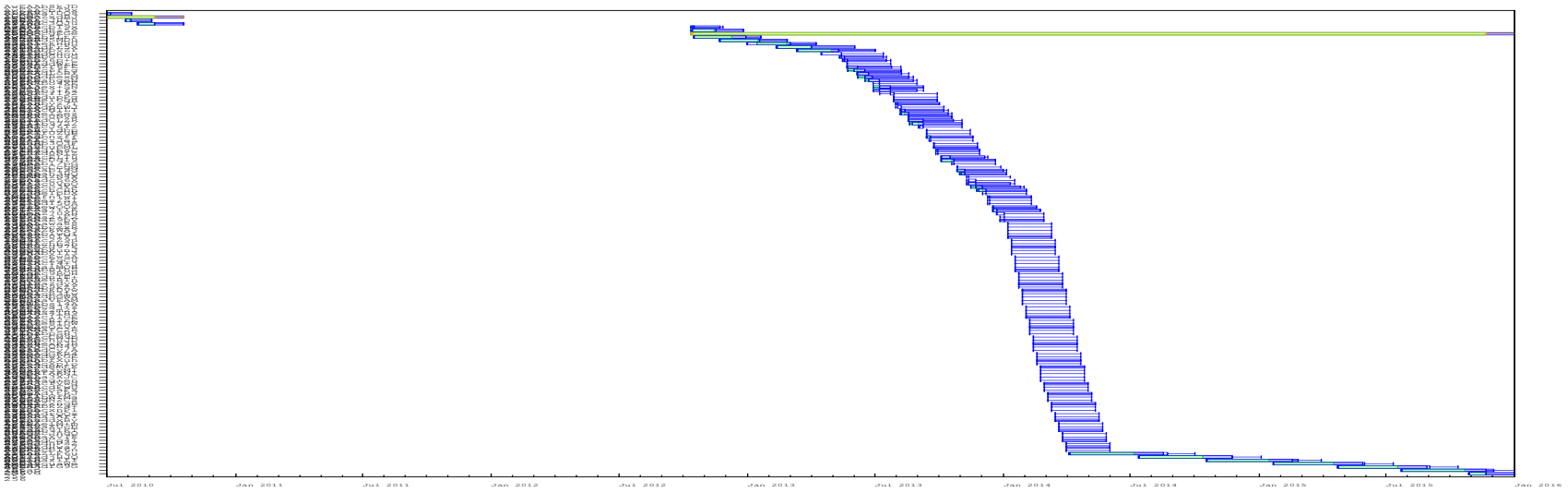
# WHAT DO ORDERLY KEY TRANSITIONS LOOK LIKE: ARIN.NET AND .COM



# THE DNS ROOT ZONE



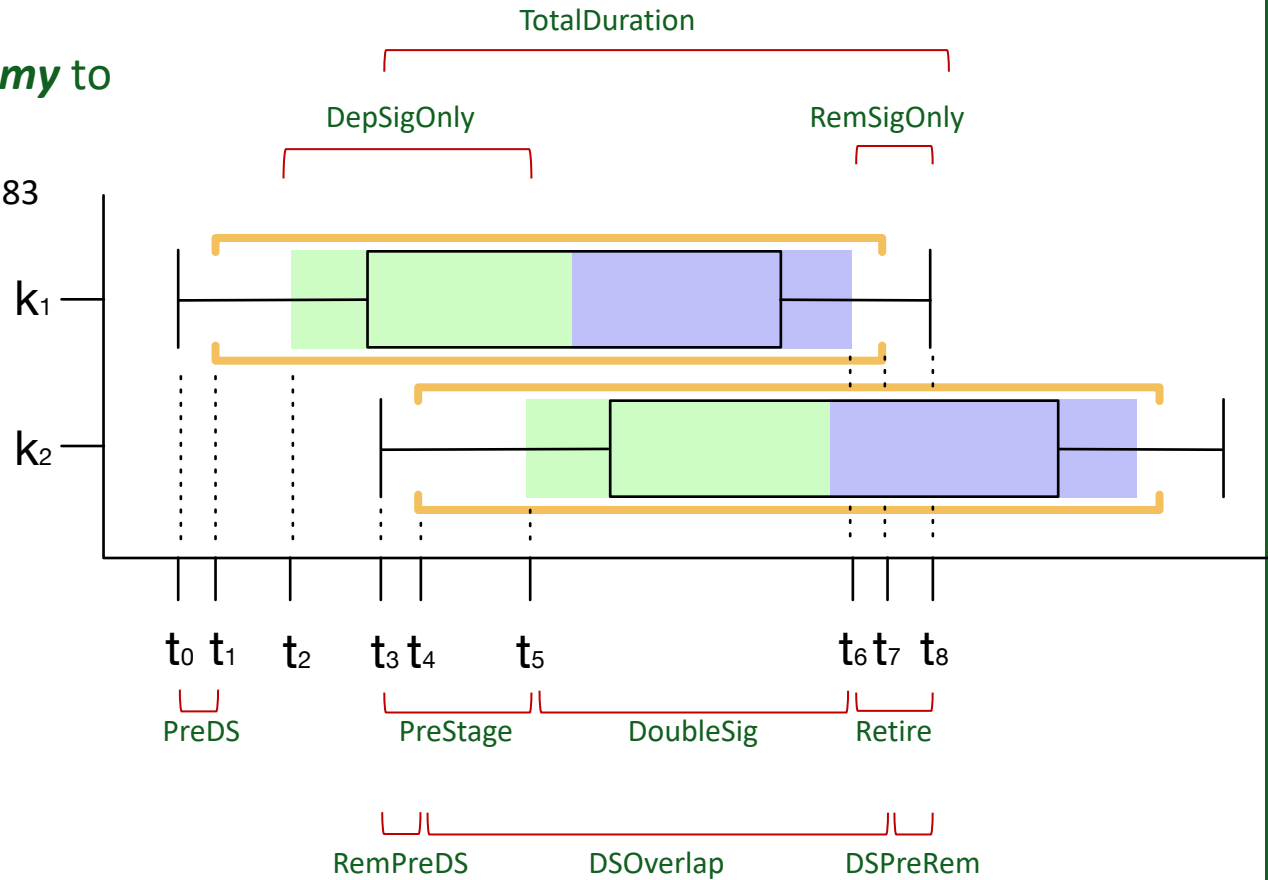
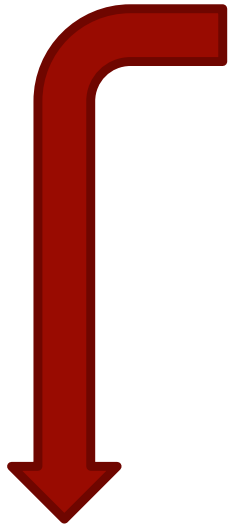
# SOME OTHER EXAMPLES



# EVALUATION: AN ANATOMY OF A KEY TRANSITION

- A key transition *anatomy* to map the topography

- Have RFCs 5011 and 7583 been used/followed successfully?



	PreDS	DoubleSig	PreStage	DepSigOnly	Retire	DSOverlap	RemSigOnly	DSPreRem	RemPreDS
ZSK Pre-Pub		= 0	> 0 M	> 0	> 0		> 0		
ZSK Double-Sig		> 0 M	= 0 M	= 0	= 0		> 0		
KSK Double-DS	< 0	= 0	= 0	= 0	= 0	> 0 M	> 0	< 0 M	< 0 M
KSK Double-KSK	> 0	> 0	= 0	= 0	> 0	= 0	> 0	> 0 M	> 0 M
KSK Double-RRset	> 0	> 0	= 0	= 0	> 0	= 0	> 0	≠ 0 M	

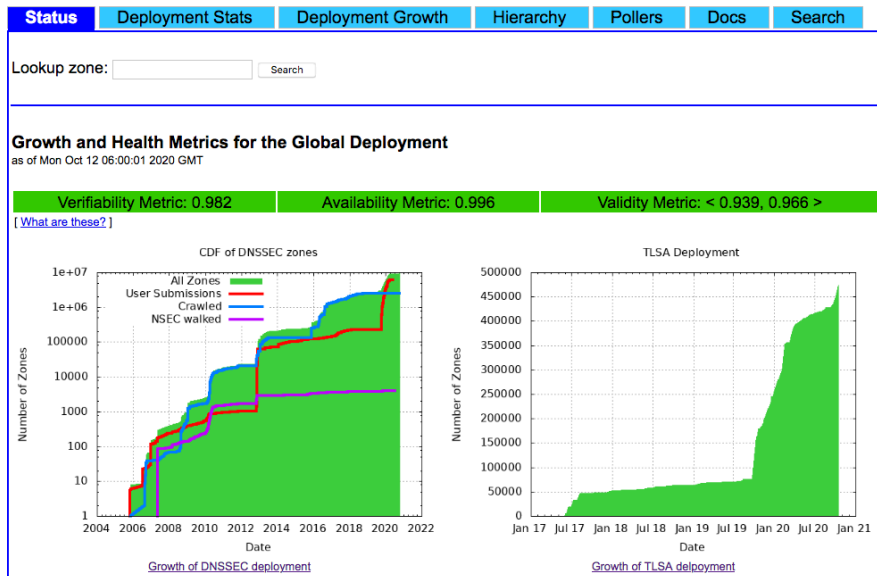
- SecSpider ( <https://secspider.net/> ): tracked DNSSEC (authoritative-side) since 2005, beginning of global rollout
- These results examine the first 10 years (2005-2015)
- 3.45 billion DNSSEC measurements
- 448,469 DNSSEC-enable zones, and 2,305,380 distinct DNSKEYs
- Now SecSpider has over 15 years of measurements, 30.8 billion rows, tracks over 7.7 million DNSSEC-enabled zones, and is still monitoring



**SECSPIDER**  
Global DNSSEC deployment tracking

Useful Links

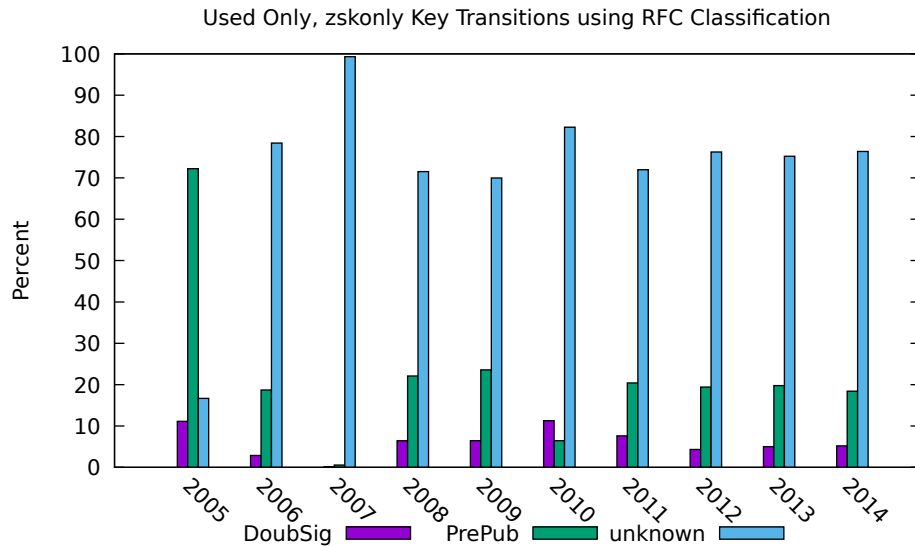
- [@SecSpider](#)
- [Why Deploy DNSSEC](#)
- [DNSSEC Deployment](#)
- [DNSSEC HOWTO](#)
- [Deploy380: DNSSEC](#)
- [DANE Info](#)
- [DANE Working Group](#)





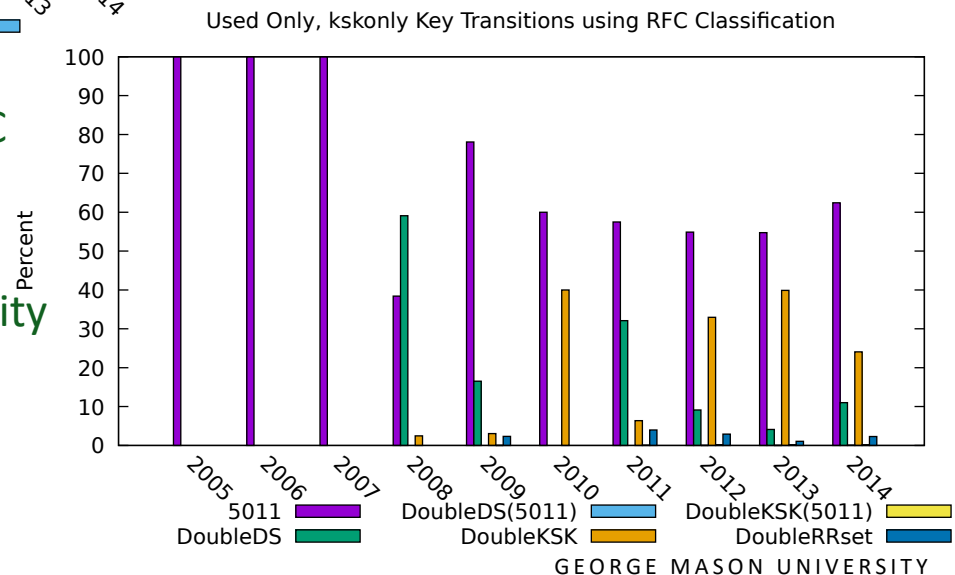
# MEASURING AGAINST THE KEY TRANSITION ANATOMY

- We measured which (if any) RFC key transition process zones followed



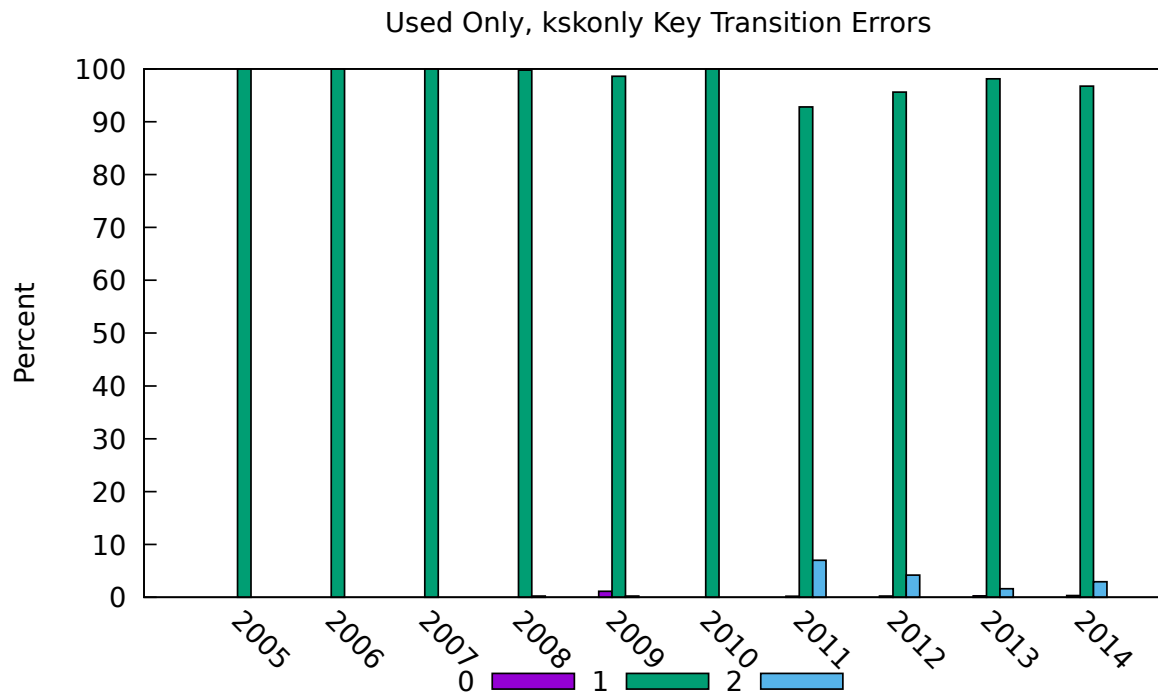
- Most ZSKs followed non-standard ZSK transitions

- For KSKs, all 5011 until the DNSSEC chain-of-trust started to develop (~2008)
- There was much more heterogeneity for KSKs



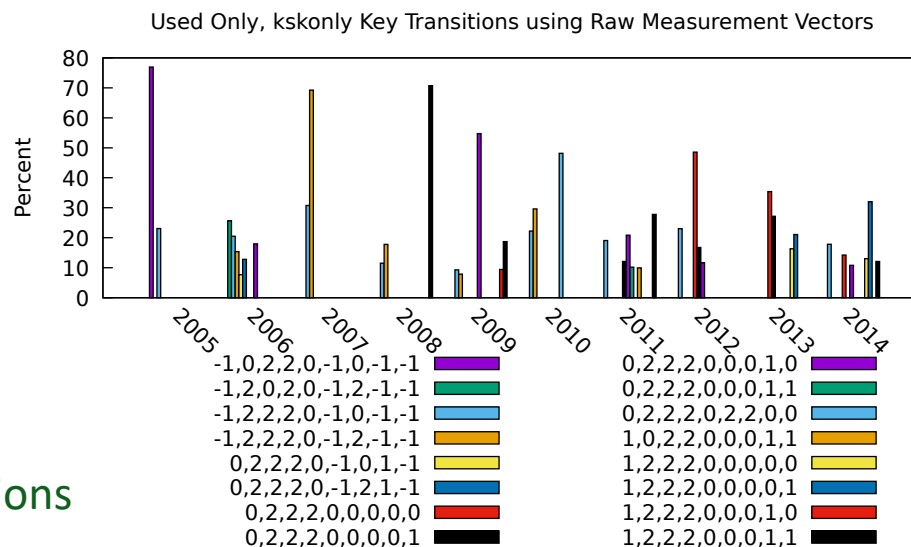
## KSK ERRORS AND WARNINGS

- For KSKs, almost all rollovers were at least in a warning state
  - 0 == no error, 1 == warning, and 2 == error
- Deviations from RFC guidance doesn't *necessarily* mean an error
  - For KSKs, only violations that affect the **correctness** of a transition constitute "error"



## DISCUSSION & FUTURE WORK

- There is a **ton more** data and results (wish I had the time to present them to you)
  - TR “DNSSEC Census: Quantifying Desire Lines in DNSKEY Transitions” posting on arXiv.org soon!
- Perhaps most exciting is to *use* the anatomy to *learn from ops*
  - Worked vs. what was standardized
- We call these “Desire Lines,” (this figure), and this is where the science will *really start!*



## FUTURE WORK

- We want to start tracking transitions in *real time*
- These analyses are just ½ the picture (auth-only)
- Will need to augment with resolving-side measurements:
  - 1+ resolvers continually issuing queries and validating the responses
  - Evaluate transition as glitch-free *iff* each query is answered correctly \*and\* DNSSEC validation always succeeds.

THANK YOU!



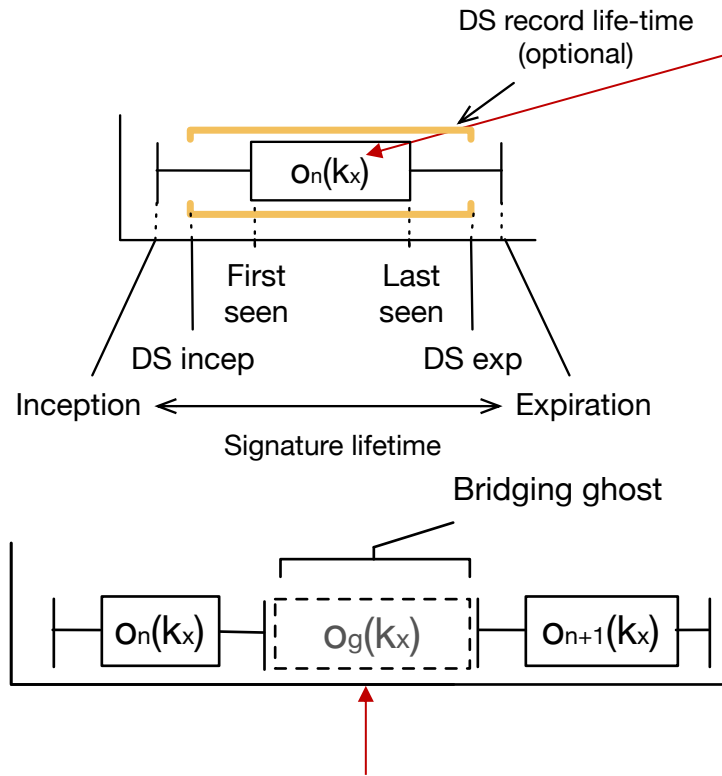
BACKUP



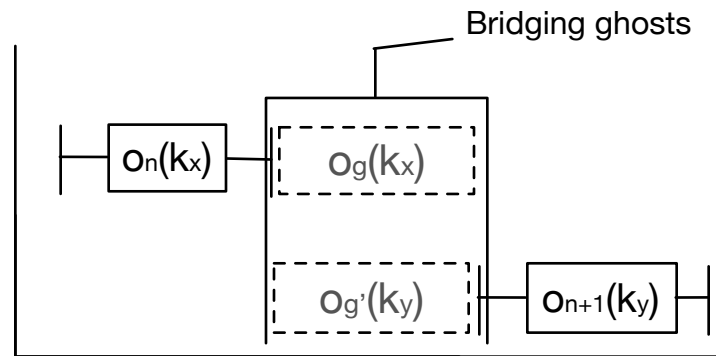
- Guidance (e.g. RFC-7583) says to periodically change the keys in DNSSEC zones
- This is commonly called “key rollover”
  - RFC guidance has prescribed ways to do this securely
  - Software tools have been implemented to make this operationally feasible
- But
  - Has it been working?
  - Have zones followed guidance?
  - Have any departures resulted in problems?
  - How would we even be able to evaluate these questions?
- This is important
  - For example, the DNS Root KSK was just rolled over for the ~~first~~  $n^{\text{th}}$  time
- To evaluate this, we can just query zones
- But, DNS resolution just gives us a snapshot of DNSKEYs served
- As keys are changed in zones, we have to examine their timing and longitudinal behaviors

# BRIDGING, BUSTING, AND BINDING

- Using longitudinal snapshots, we can Bridge, Bust, and Bind instantaneous measurements into continuous models
- We created a novel technique called *Bridging, Busting, and Binding*



DNSKEY observation:  $O_n(k_x)$



Ghosts are "**busted**" if they can be refuted, don't make sense, etc.

Ghost "**bridges**" the gap between observations/measurement

## GREAT... BUT WHAT DO THESE PICTURES *MEAN*???

- Have these processes been “working?”
- Have zones followed guidance?
- Have any departures resulted in problems?
- What are the differences between these processes for KSKs and ZSKs?
- How would we even be able to evaluate these questions?



## FIRST, WHAT IS A KEY ROLLOVER?

- Is it whenever an (old) key gets *securely* replaced by a (newer) key?
- Are “rollovers” when a single key changes to another key (a *1:1* transition)
- Then, if there are  $n$  keys, and a zone transitions to  $m$  keys, which key(s) rolled over to which other keys?
  - Did all disappearing keys rollover to each/all of the remaining keys?
  - If some other keys remained, did they get rolled over to as well?
- The word “rollover” is *not* expressive enough when  $n$  keys transitions to  $m$  keys ( $n : m$ )
- We propose “key transition” is superset of key rollovers
  - That is, a degenerative case of an  $n:m$  transition may be a *1:1* rollover

## BUT WE STILL NEED MORE

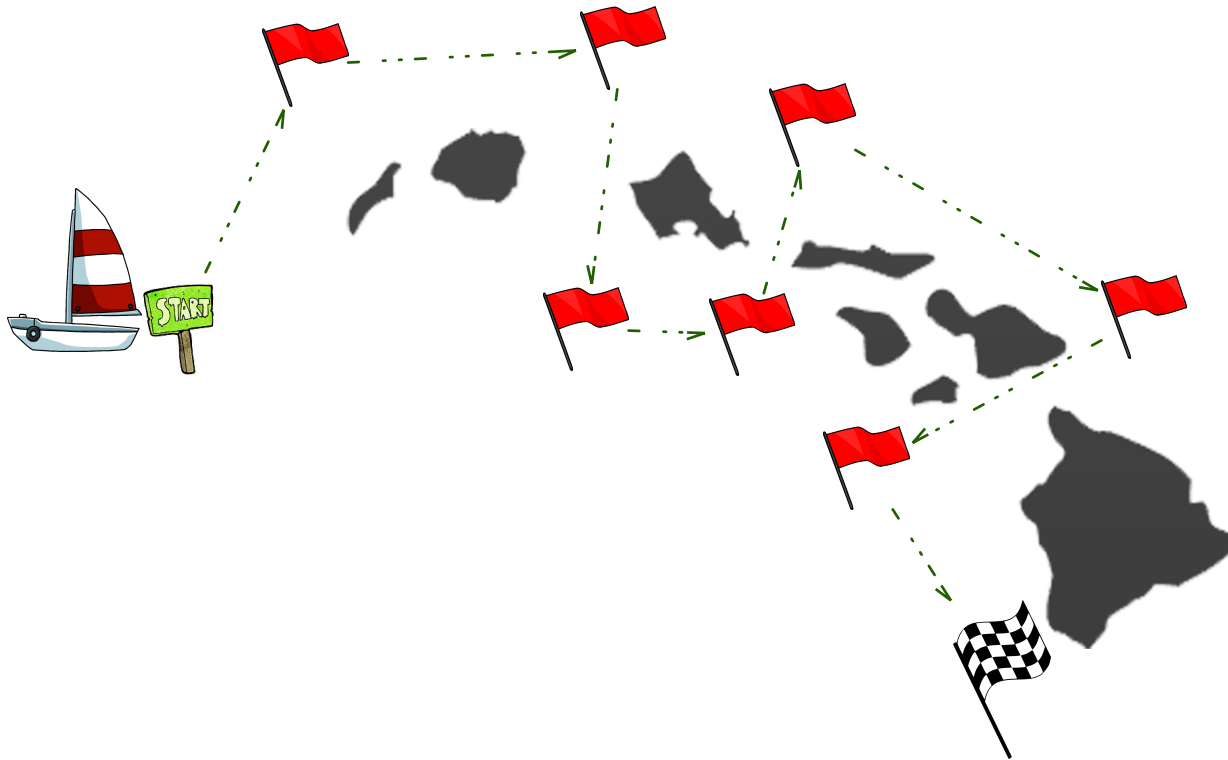
- Need a structured way to **evaluate** whether key transitions...
  - Are following guidance
  - Whether there are process warnings or errors that are leading to security concerns
  - Whether there are non-standardized behaviors that may actually be **optimizations**
  - etc.
- Our approach is to map out a **topography** of key transitions
- We have to know
  - What to measure
  - What is meaningful
  - What results actually mean
  - etc.
- Analogy: how could we evaluate boats racing in a regatta?
  - Do they follow the course arcs?
  - Are there collisions? ←
  - etc.

No, not “name collisions!!”  
Too soon?

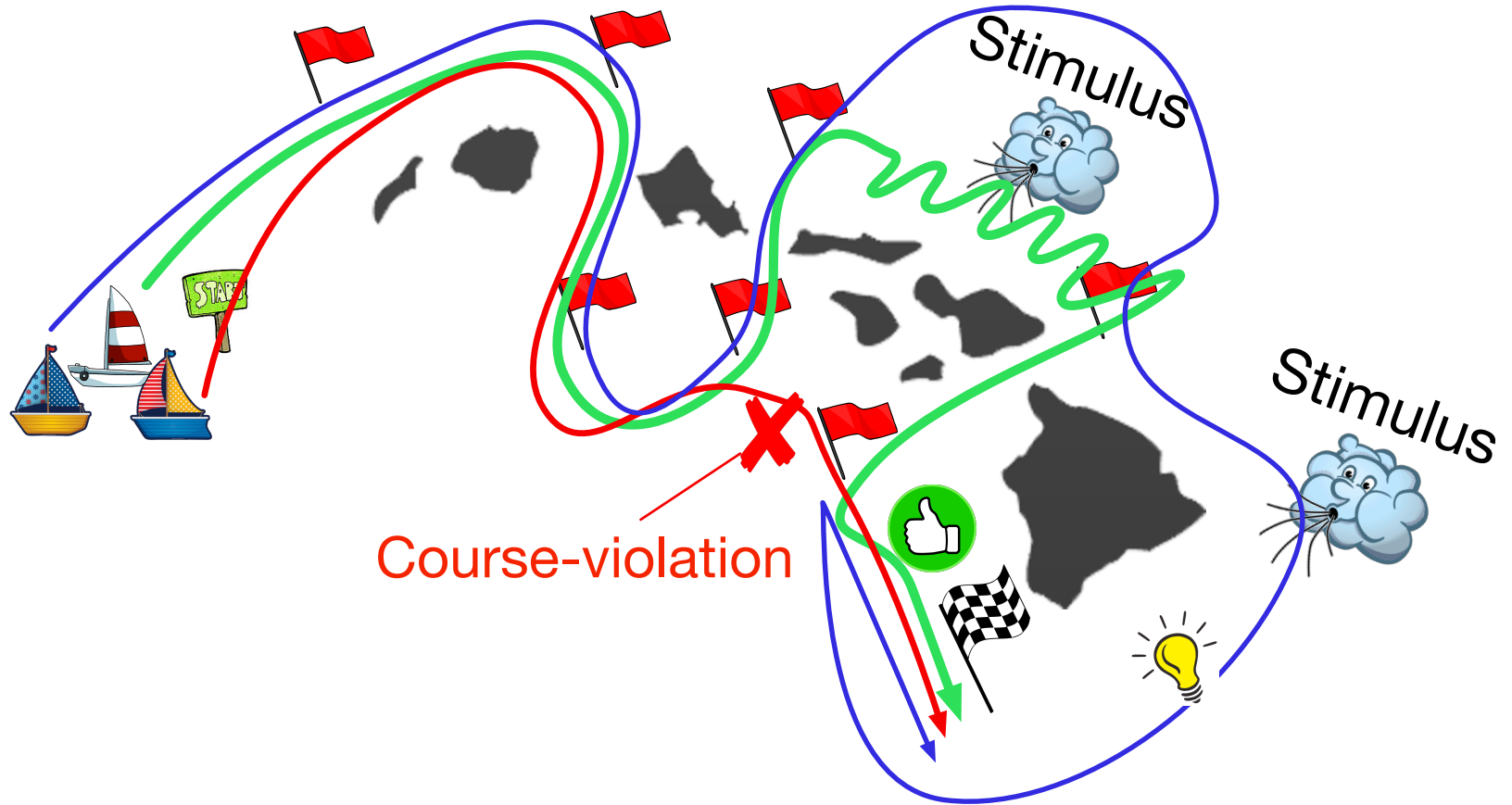
## APPLYING THIS TO DNSSEC

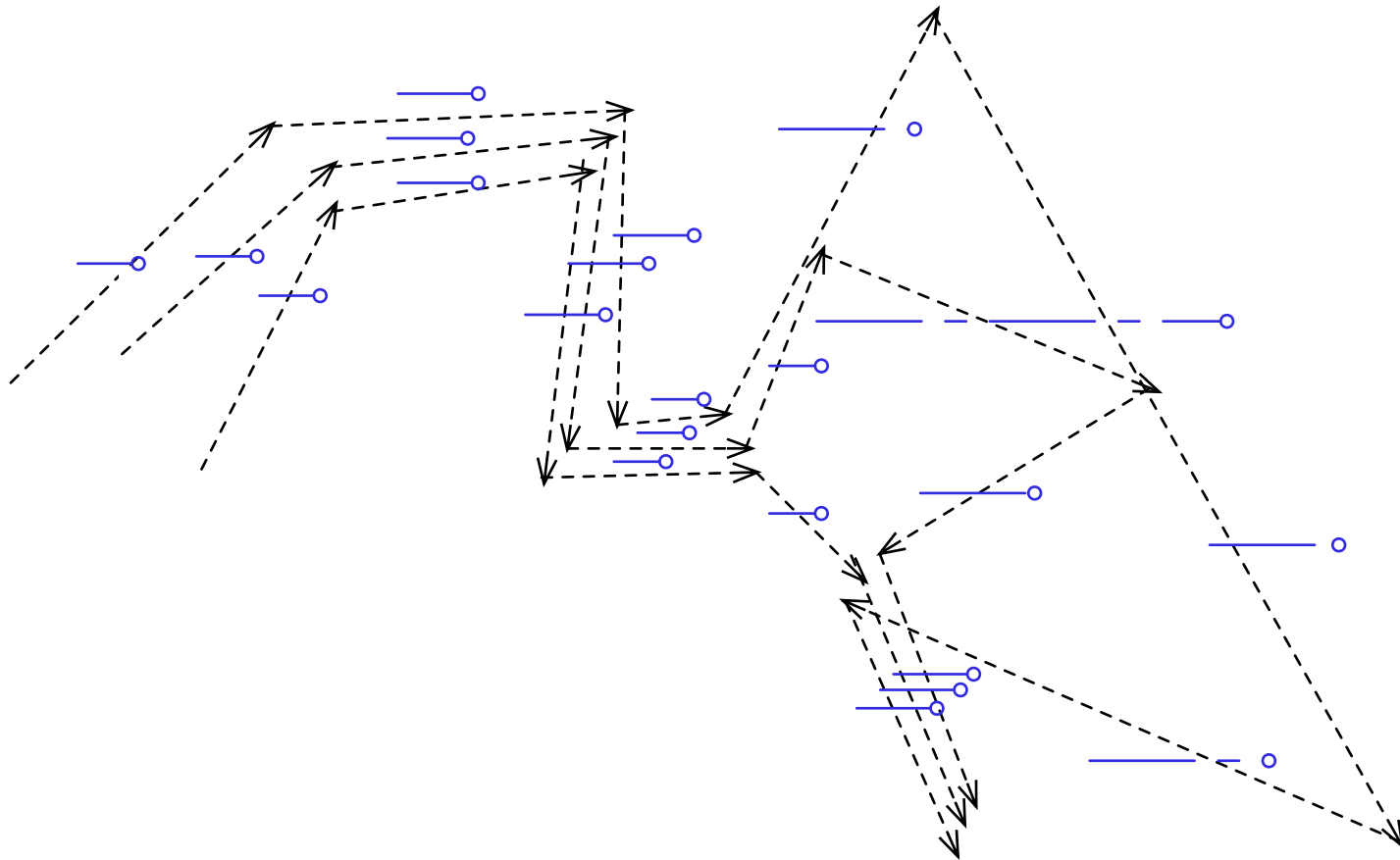
- To quantify DNSKEY transitions, what should be the analogs to regatta “way points”?
- We developed an ***anatomy*** of DNSSEC key transitions to let us concisely measure and evaluate how transitions are effectuated
- Our anatomy is designed to inform *what* we need to measure, and why

# REGATTA EXAMPLE COURSE



# HOW CAN WE QUANTIFY/EVALUATE/DETECT BEHAVIORS?





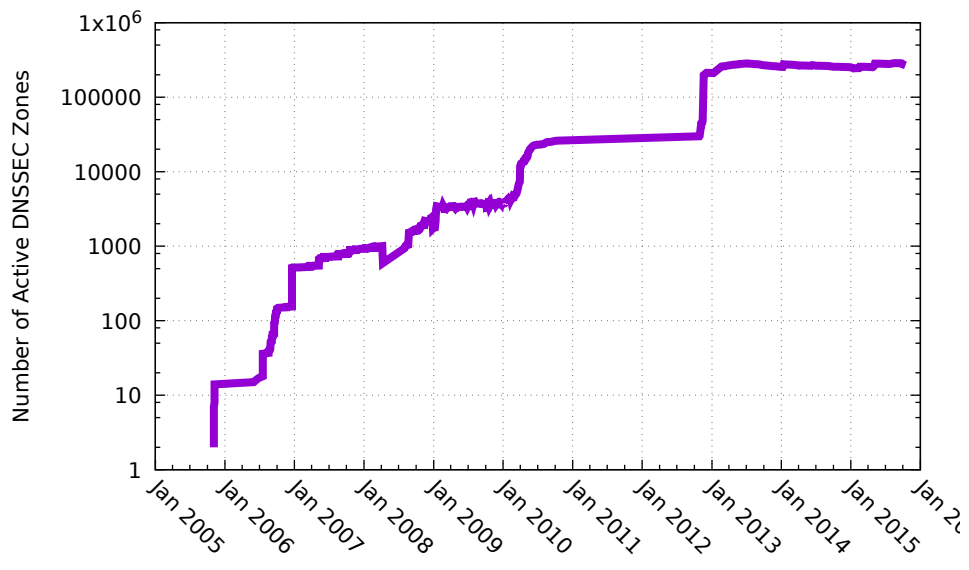
- Turning a boat's continuous course into (for example) piecewise linear components lets us quantify its journey

PreDS	If departing key covered by DS, duration it was verifiable before DS(es) (Note: can be negative)
DoubleSig	Duration that both removed key and remaining key were used for verifying zone data
PreStage	The amount of time that the remaining key was valid, but before being used to verify zone data
DepSigOnly	The duration during the key transition when only the departing key was in use
Retire	Amount of time departing key was still valid but after it was no longer in use
DSOverlap	The duration (if at all) that DS(es) for the departing and remaining keys overlapped
RemSigOnly	The duration during the transition when only the remaining key was usable to verify signatures
DSPreRem	If departing key covered by DS, the amount of time it was valid after DS(es) gone (can be negative)
RemPreDS	If the remaining key is covered by a DS, the duration that it was verifiable before its DS(es)
TotalDuration	The duration of the entire key transition

- With this anatomy, we can *quantify* RFC guidance!
- We can discretize measurements as  $>0$ ,  $=0$ ,  $<0$ , or N/A
- We have started with RFCs 5011 and 7583:

	<i>PreDS</i>	<i>DoubleSig</i>	<i>PreStage</i>	<i>DepSigOnly</i>	<i>Retire</i>	<i>DSOverlap</i>	<i>RemSigOnly</i>	<i>DSPreRem</i>	<i>RemPreDS</i>
<i>ZSK Pre-Pub</i>		= 0	> 0 M	> 0	> 0		> 0		
<i>ZSK Double-Sig</i>		> 0 M	= 0 M	= 0	= 0		> 0		
<i>KSK Double-DS</i>	< 0	= 0	= 0	= 0	= 0	> 0 M	> 0	< 0 M	< 0 M
<i>KSK Double-KSK</i>	> 0	> 0	= 0	= 0	> 0	= 0	> 0	> 0 M	> 0 M
<i>KSK Double-RRset</i>	> 0	> 0	= 0	= 0	> 0	= 0	> 0	≠ 0 M	



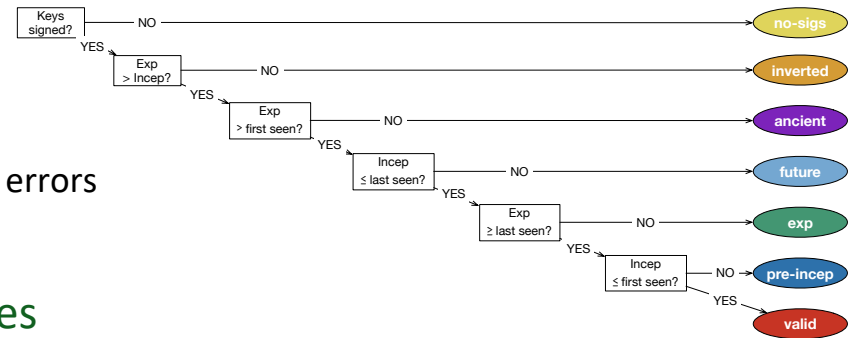


# KEY LIFE CYCLE MANAGEMENT

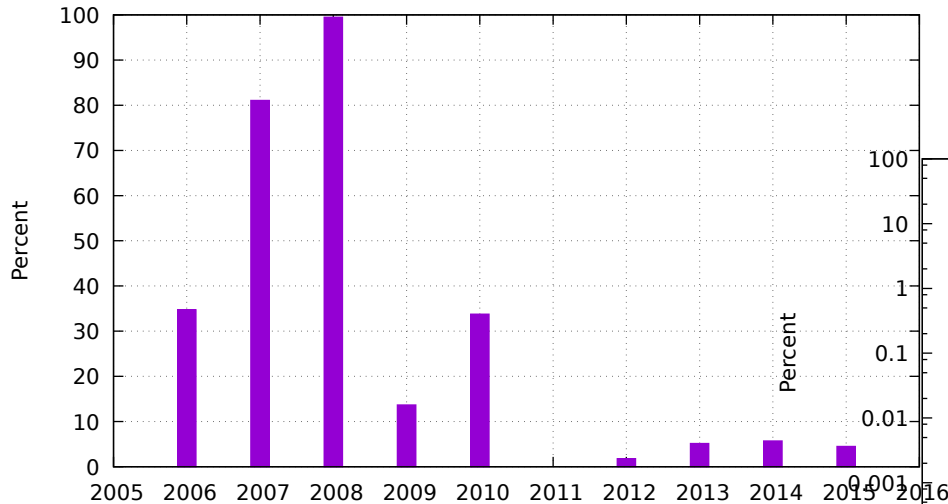
- Created classifier

- Quantify certain error states
- Valid is classified as the absence of classified errors

- Calculated rates of errors and their types



Yearly Rate of State Errors



Key States

