DNSSEC Key Management Policy

Edward Lewis

Neustar

DNSSEC JP!

Agenda

- What is Key Management?
- Why and Where it fits?
- Key Management in detail
- Our experience in "dotUS" and "dotBIZ"

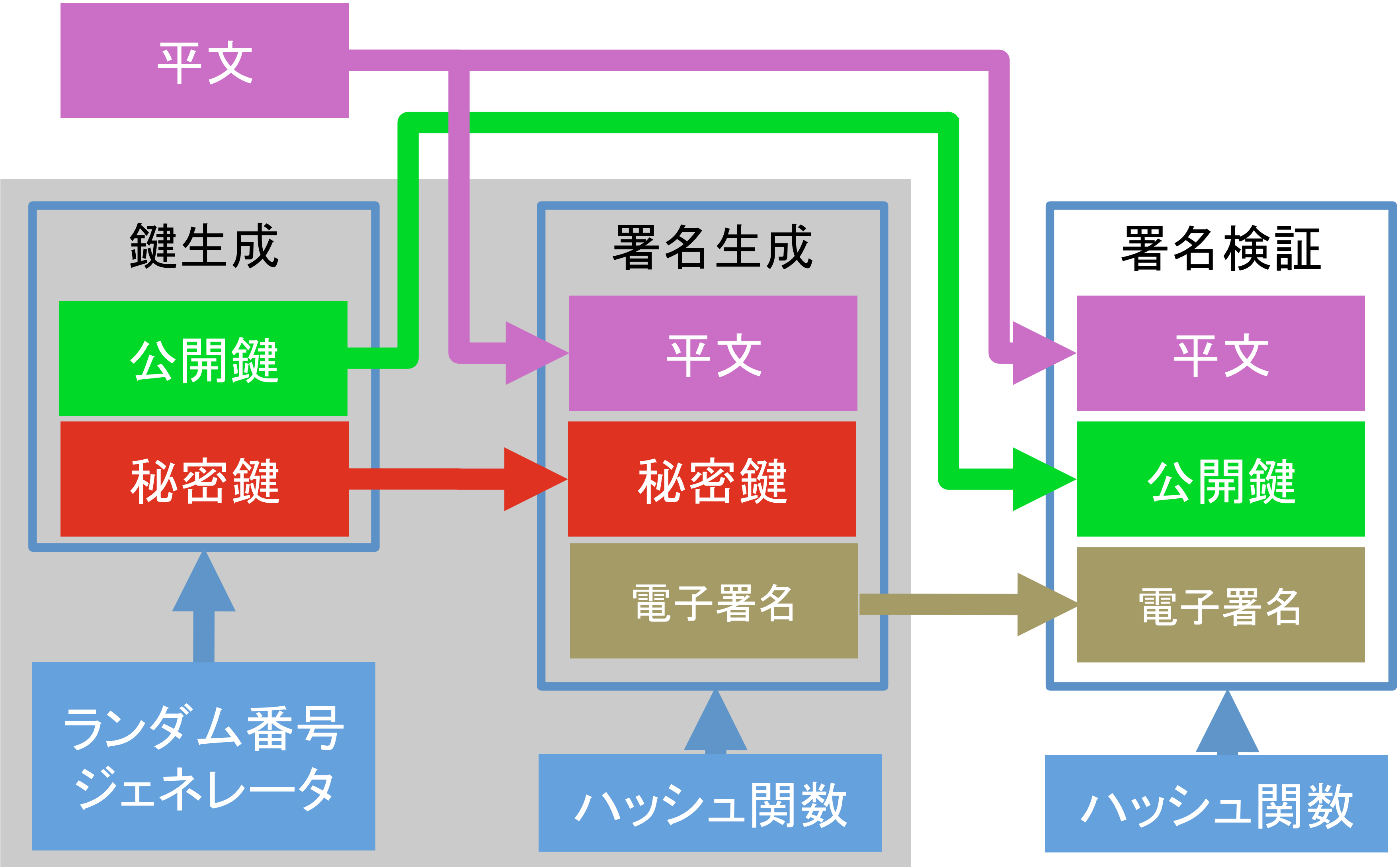
Key Management

- I learned a lot by reading US NIST documents
  - I am not sure if the same exist in Japan
- A reading list for Key Management & DNSSEC
  - http://csrc.nist.gov/publications/PubsSPs.html
  - SP800-57, also see SP800-53, SP800-81
- Helpful information on HSM devices
  - http://csrc.nist.gov/publications/PubsFIPS.html
  - FIPS 140-2

Why Manage Keys?

- DNSSEC uses keys to produce digital signatures
- Keys are used in different ways
- Keys have "lifetimes" and cannot be considered to be "forever"
- There is a lot of debate on how long to use a key and how a key should be used

電子署名の概念


平文
鍵生成
公開鍵
秘密鍵
ランダム番号ジェネレータ
署名生成
平文
秘密鍵
電子署名
ハッシュ関数
署名検証
平文
公開鍵
電子署名
ハッシュ関数

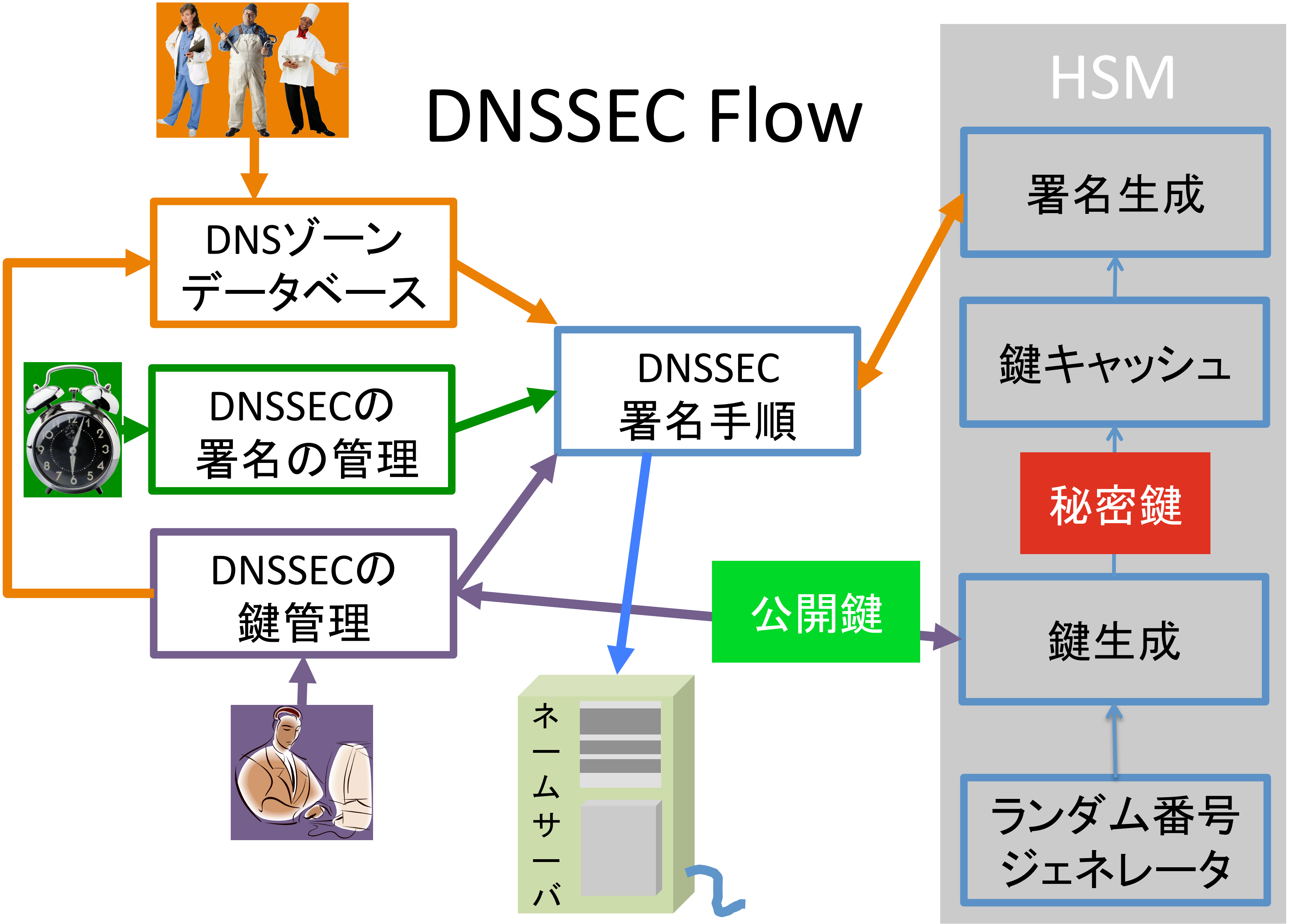
The grey box

- On the previous slide a gray box groups
  - 鍵生成
  - 署名生成
  - ランダム番号ジェネレータ
  - ハッシュ関数
- These functions may be in a software library (like OpenSSL) or may be in an Hardware Security Module (HSM)

3 ways data changes in DNSSEC

- Zone file's data changes (same in DNS)
  - New hosts/addresses/etc.
- Signatures expire (*new* in DNSSEC)
  - DNSSEC relies on expiration time for revocation
  - Signatures have to be "refreshed"
- Key/cryptographic material changes (also *new*)
  - Keys and algorithms don't last forever
  - Recovery from an attack may require new keys

DNSSEC Flow



DNSゾーン データベース

DNSSECの 署名の管理

DNSSECの 鍵管理

DNSSEC 署名手順

ネームサーバ

公開鍵

HSM

署名生成

鍵キャッシュ

秘密鍵

鍵生成

ランダム番号 ジェネレータ

HSM &DNSSEC署名手順

- HSM or software cryptographic library
  - Provides the "mathematic muscle" for cryptography
  - (non-HSM: Openssl libraries)
- DNSSEC署名手順 signs current data with current keys
  - Puts the cryptography into DNSSEC records, zones
  - Feeds the name server

When and How (to Sign) Policy

- Decision of when to sign is governed by
  - DNSゾーンデータベース because data is changed
  - DNSSECの署名の管理 because signatures are expiring (or wall-clock alarm strikes)

- Decision of what key to use is governed by
  - DNSSECの鍵管理 has to manage the current set and changes to the next set of keys

Please Recycle

- Regarding "when to sign"
  - Generating new signatures before it is necessary to do so is discouraged
  - Zone transfers become large and name servers still are not good at juggling zone transfers and queries

- If a zone is static, let signatures live long and refresh them with short overlaps

DNSゾーンデータベース

- Changes to the zone contents will cause DNSSEC signing to happen
  - User changes to the zone (new host)
  - DNSSECの鍵管理 delivers a new 公開鍵
  - (Not shown) NSEC3 parameter is changed
- Policy
  - A zone must always have a complete set of fresh signatures. No exceptions!

DNSSECの署名の管理

- DNSSEC signatures have expiration times
  - When a signature expires it must be refreshed
  - Usually this function is built into other tools
- Policy
  - Rule of thumb, refresh signature well before expiration to give enough time to "recover" from a failure

DNSSECの鍵管理

- Determines if the existing keys are "good" or if there is a need to change
- Key Management Policy
  - Following slides
- Policy implementation
  - Requires new key pairs to be generated
  - Sends DNSゾーンデータベース new public keys
  - Rotates keys into and out of service
  - Revokes keys

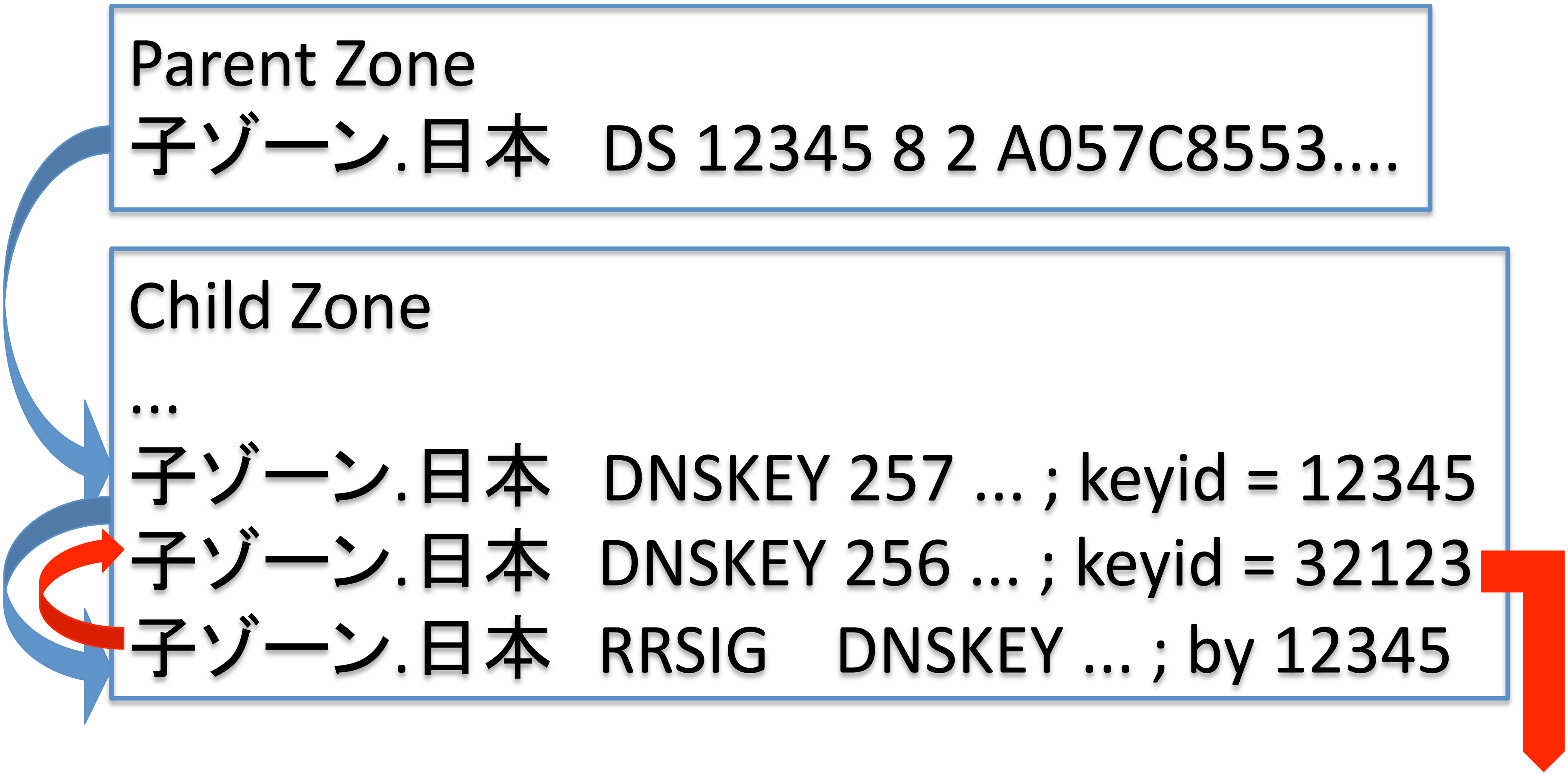
Key Management Policy Aspects

- Key roles
  - Use KSK/ZSK or not? Follow RFC 5011?
- Key algorithm (and hash) and size
  - RSA SHA256? SHA1? SHA512? GOST?
  - 1024 bits or 2048 bits?
- Key lifetime
  - Duration of key "effectivity" period
  - Procedure and timing of key change

Key Roles

- Choose KSK/ZSK or just one key?
  - If the parent zone is fast and responsive, one key is good
  - But if the parent is slow, the KSK/ZSK approach is worth the management of the extra key
- KSK/ZSK
  - Assumed by DNSSEC early adopters, not a requirement
  - See RFC 4641

KSK/ZSK



Parent Zone

子ゾーン.日本 DS 12345 8 2 A057C8553....

Child Zone

...

子ゾーン.日本 DNSKEY 257 ... ; keyid = 12345

子ゾーン.日本 DNSKEY 256 ... ; keyid = 32123

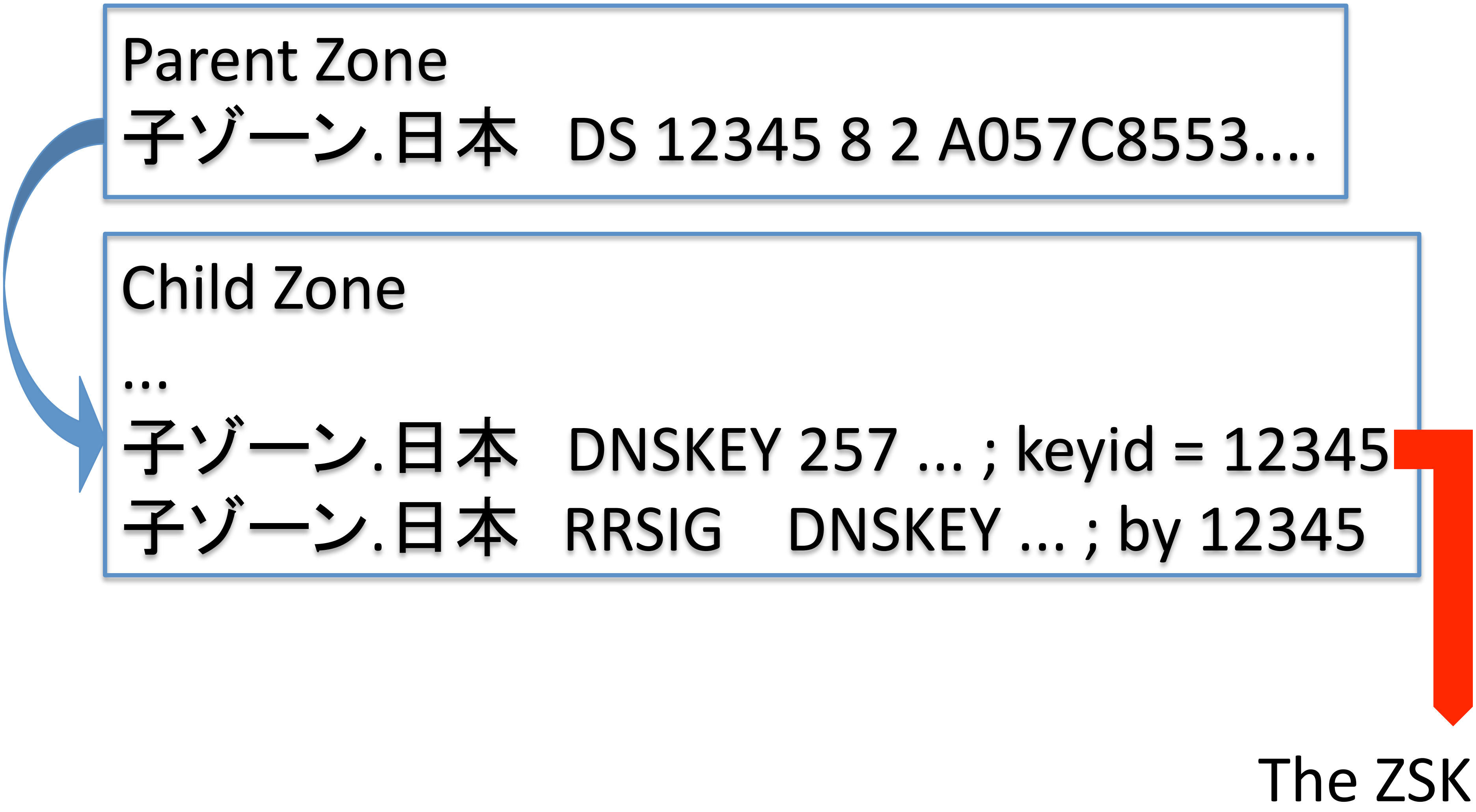
子ゾーン.日本 RRSIG DNSKEY ... ; by 12345

The ZSK

Single Key DNSSEC

- Managing 1 key is simpler than managing 2
- But only if you have a "quick" relationship with your parent zone
  - Need to change the DS record every time you change the key signing the zone
  - Or, if you never change keys...
- Since the invention of EPP, this is plausible
  - You can try it, but I still encourage KSK/ZSK

Single Key "Chain"



Parent Zone

子ゾーン.日本 DS 12345 8 2 A057C8553....

Child Zone

...

子ゾーン.日本 DNSKEY 257 ... ; keyid = 12345

子ゾーン.日本 RRSIG DNSKEY ... ; by 12345

The ZSK

RFC 5011

- Management of trust anchors
  - A new key has to be present for some time to verify it is indeed a new key
  - A revoked key is marked and signed for some time to verify the key is removed
- Intended for use where the parent zone is not signed or won't hold DS records

Key Algorithm and Size

- DSA, RSA, RSA+NSEC3, GOST
  - See http://www.iana.org/assignments/dns-sec-alg-numbers/dns-sec-alg-numbers.xhtml
- Hash function
  - SHA-1 , SHA-256 or something else?
  - SHA-1 is considered to be "old" but still in use
- Size
  - Longer is harder to break, slower to use

The Hash Function

- SHA 1
  - Published in 1995
  - 160 bits
  - Widespread, but getting to be "breakable"
- SHA 2 (or SHA 256 or SHA 512)
  - Published in 2001
  - 224/256 or 384/512 bits
  - More bits, harder to "break"

Is longer better and slower?

- A longer key is thought to be
  - harder to "crack" so it is more secure
  - harder to process so it is less efficient
- What do cryptographers feel?
  - DNSSEC is uses a subset of cryptographic functions
  - There isn't enough use of a key to crack it, provided it is strong enough (1024 bits)
- Frankly, no one has enough experience yet

Key Lifetime

- Lifetime, from creation to deletion, comprises
  - Key effectivity period, the duration a key is used cryptographically
  - Key DNSSEC lifetime, the durations needed to publish and remove a key, DNS TTL plays a role
  - RFC 5011 impacts timing to allow detection of key changes if there is no parent signing

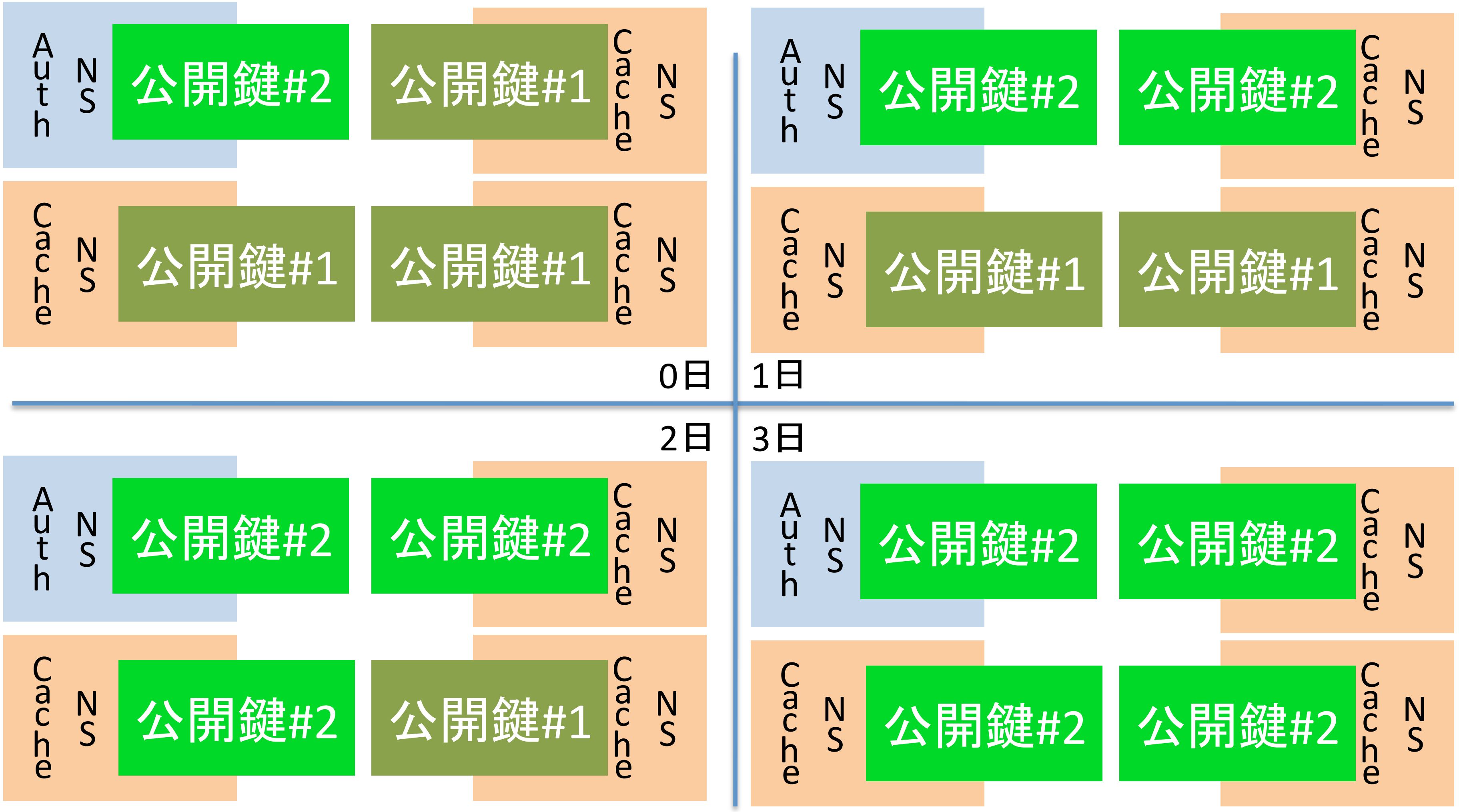
Key effectivity period

- There is some debate
  - DNSSEC developers thought that keys had to be changed because of cryptographic properties
  - Cryptographers have said (opinion) that keys will be good "until broken" (which is true)
  - In operations, regular changes are good because
    - Broken keys may not be detected
    - Keys cannot be revoked (RFC5011 is a special case)
    - Operational scripts need to be exercised

TTL impacts

- http://tools.ietf.org/html/draft-morris-dnsop-dnssec-key-timing-02
- Assume a key is effective for 3 months
- What about DNS zone and cache propagation?
  - A new key has to be pre-published to avoid a cache with a "new data signature and old keys."
  - An old key has to hand around until all of its signatures are gone

Cache Impact



0日

Auth NS 公開鍵#2 — 公開鍵#1 Cache NS

Cache NS 公開鍵#1 — 公開鍵#1 Cache NS

1日

Auth NS 公開鍵#2 — 公開鍵#2 Cache NS

Cache NS 公開鍵#1 — 公開鍵#1 Cache NS

2日

Auth NS 公開鍵#2 — 公開鍵#2 Cache NS

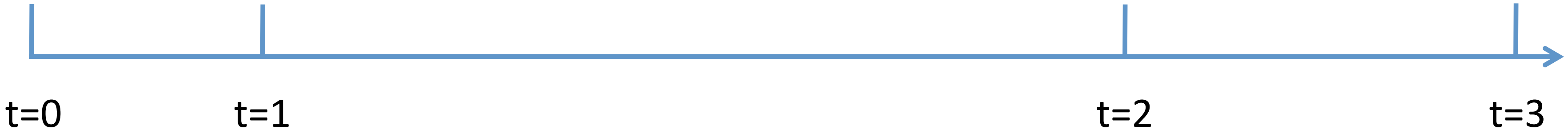
Cache NS 公開鍵#2 — 公開鍵#1 Cache NS

3日

Auth NS 公開鍵#2 — 公開鍵#2 Cache NS

Cache NS 公開鍵#2 — 公開鍵#2 Cache NS

DNSSEC Basic DNSKEY cycle



- t=0 DNSKEY is added to zone
- until t=1 Some caches will have the old set
- t=1 All caches should have DNSKEY
- until t=2 Private key can make RRSIG
- t=2 private key retired
- until t=3 RRSIGs in Caches, DNSKEY needed
- t=3 DNSKEY is removed from zone

BIND key management

- In BIND 9.7 there is a new key mangement feature
  - (P)ublish is t=0
  - (A)ctivate is t=1
  - (I)nactivate is t=2
  - (D)elete is t=3

Experience in US and BIZ

- US signed in December 2009, open for DS records in June 2010
- BIZ began signing July 2010

- Both zones are using NSEC because there is no reason to use NSEC3
  - Zones can be retrieved via FTP
  - We aren't concerned about size

My personal TLD survey

- I have a script that asks for DNSKEY from the delegations in the root and in ARPA
  - Skewed by test zones in the root
  - ARPA includes e164.ARPA and other signed zones
- As of early July, 24 "real" TLDs are signed
  - I use this only for sanity checking, not reliable as a measure of overall DNSSEC adoption
  - You will see reference to "41" - that includes test zones and ARPA zones

Key Roles

- We use KSK/ZSK
  - Because our parent is slow (the root), no automatic update interface and no quick turnaround
  - We plan to change keys frequently
- Survey, 40 out of 41 use KSK/ZSK
  - But that isn't surprising as we all think alike
- Single key use
  - Workable but in my opinion, not too scaleable

Key Algorithm

- No crypto system is imposed (by law) so we choose what seems best
- From the 41 signed zones in the root plus ARPA all use RSA
  - 9 zones use RSA-SHA256, rest use RSA-SHA1
- Recommendation
  - Unless you must use an algorithm for legal reasons, choose RSA-SHA256
- Don't *start* with RSA-SHA1 (NSEC or NSEC3!)

Key Sizes

- We have stuck to the common wisdom of a KSK of 2048 and a ZSK of 1024 bits
- Survey "the most common set up"

Role	Hash	"NSEC"	1K	2K	4K	Odd
KSK	SHA1	NSEC	0	21	2	1
ZSK	SHA1	NSEC	23	0	0	1
KSK	SHA1	NSEC3	2	5	0	
ZSK	SHA1	NSEC3	7	1	0	
KSK	SHA256	–	0	8	0	
ZSK	SHA256	–	8	0	0	

Key Lifetime

- Key effectivity
  - 1K bit ZSK - 3 months
  - 2K bit KSK - 1 year
- Our parameters
  - ZSK published as a emergency key for 3 months, signs for 3 months
  - KSK is published for 1 year as the emergency and 1 year as the active (DS at root)
  - TTL is 6 days

RFC 5011 support

- We plan to support RFC 5011
  - But in reality we could just rely on the root zone to have the DS record
  - As a safety mechanism, we publish our key set on a website, so RFC 5011 support is a good thing
- No clear recommendation on RFC 5011
  - Needed if parent is not signed
  - Probably not if signed

Questions

- Questions?