

2010年、今そこにある課題
「IPv6対応とDNSSEC」
DNSSEC編

民田雅人
株式会社日本レジストリサービス
HOSTING-PRO 2010 W1

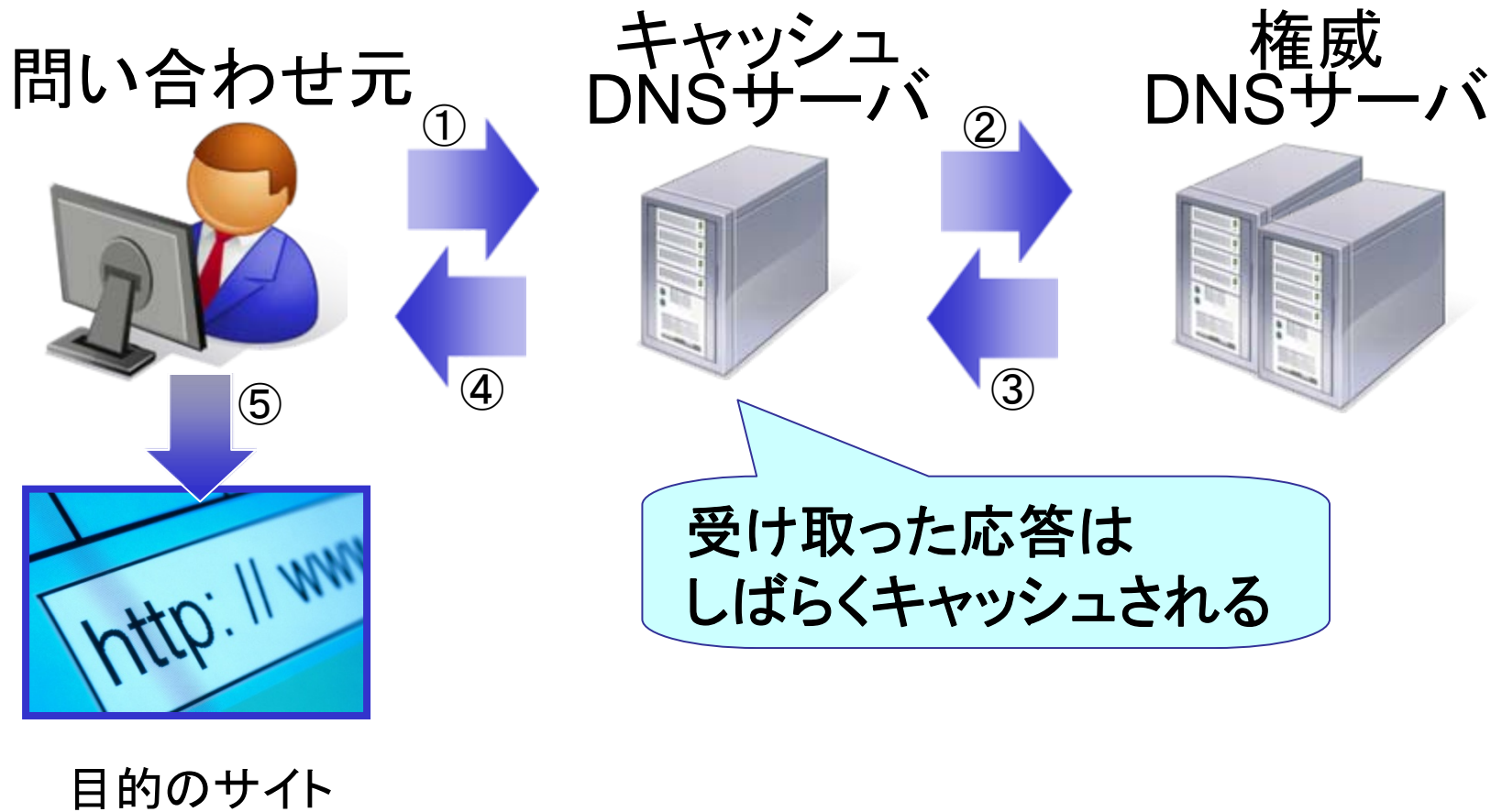
DNSキャッシュへの毒入れと DNSSEC

DNSへの毒入れ (キャッシュポイズニング)

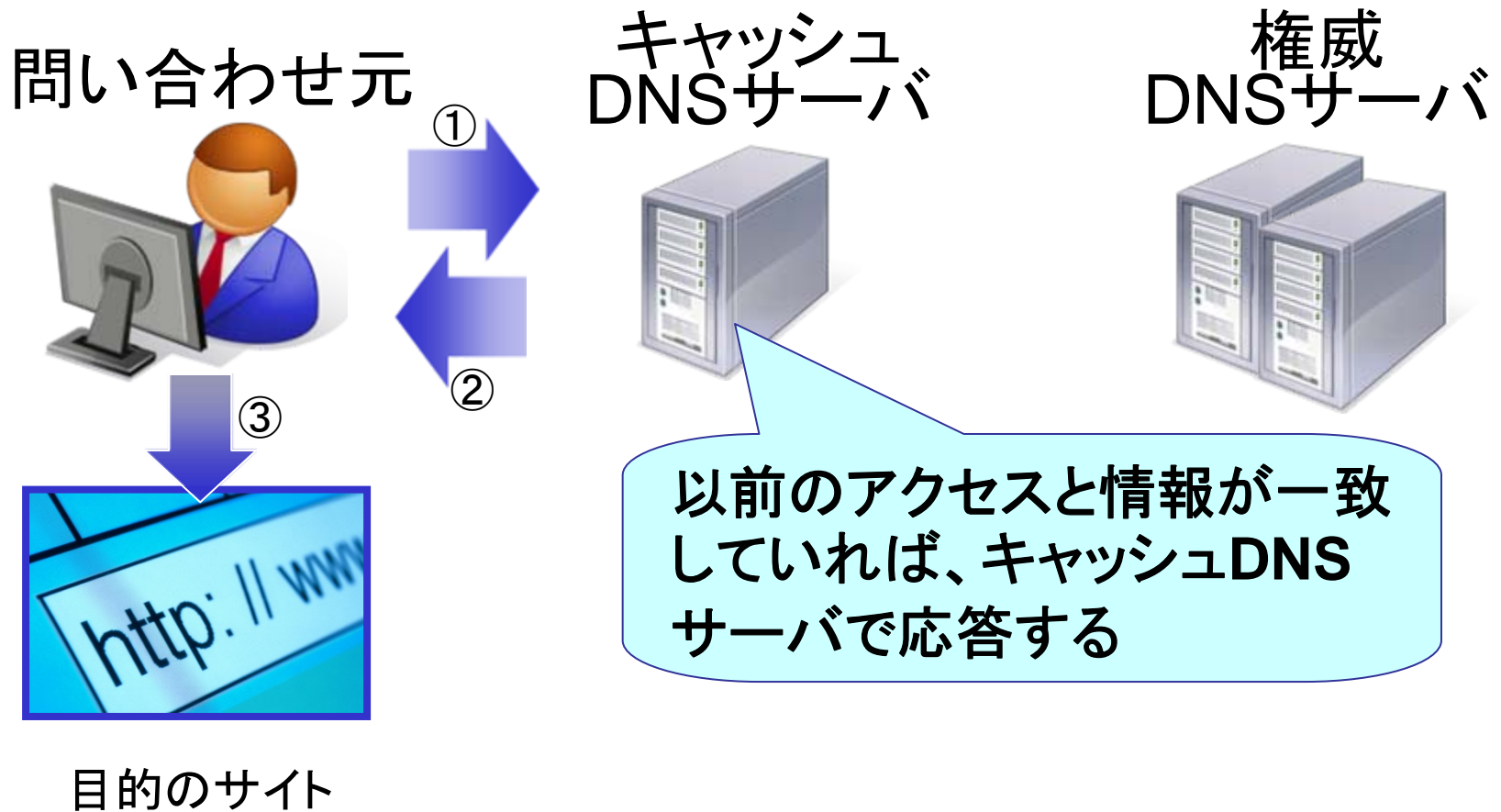
- 予めキャッシュDNSサーバに偽の情報を覚えこませ、ユーザが正しいアクセスを行ったつもりでも、偽装サイトへ誘導する手法
– フィッシングの為の攻撃手法の一つ

DNS最大級のリスク

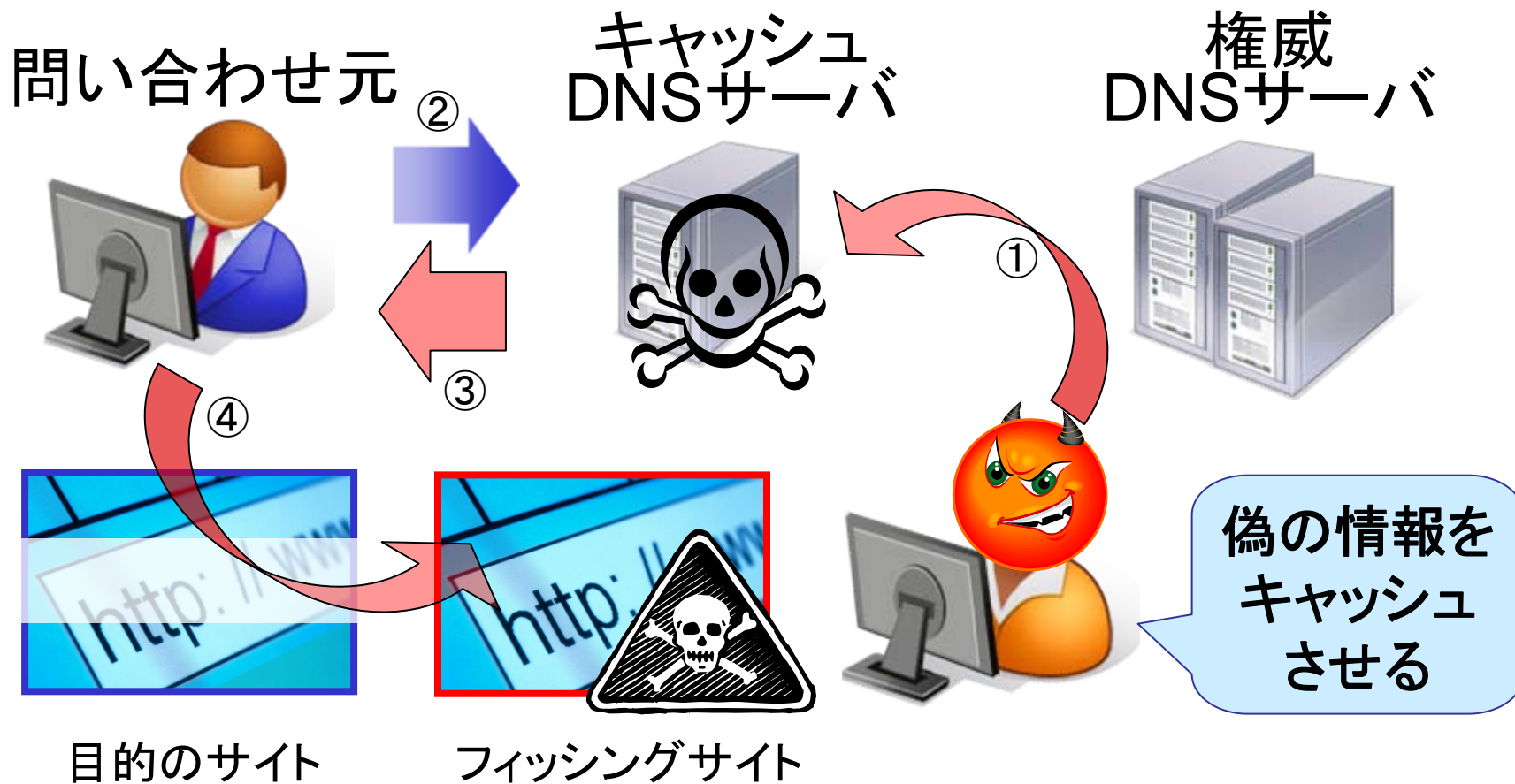
DNSの正常な流れ(1回目のアクセス)



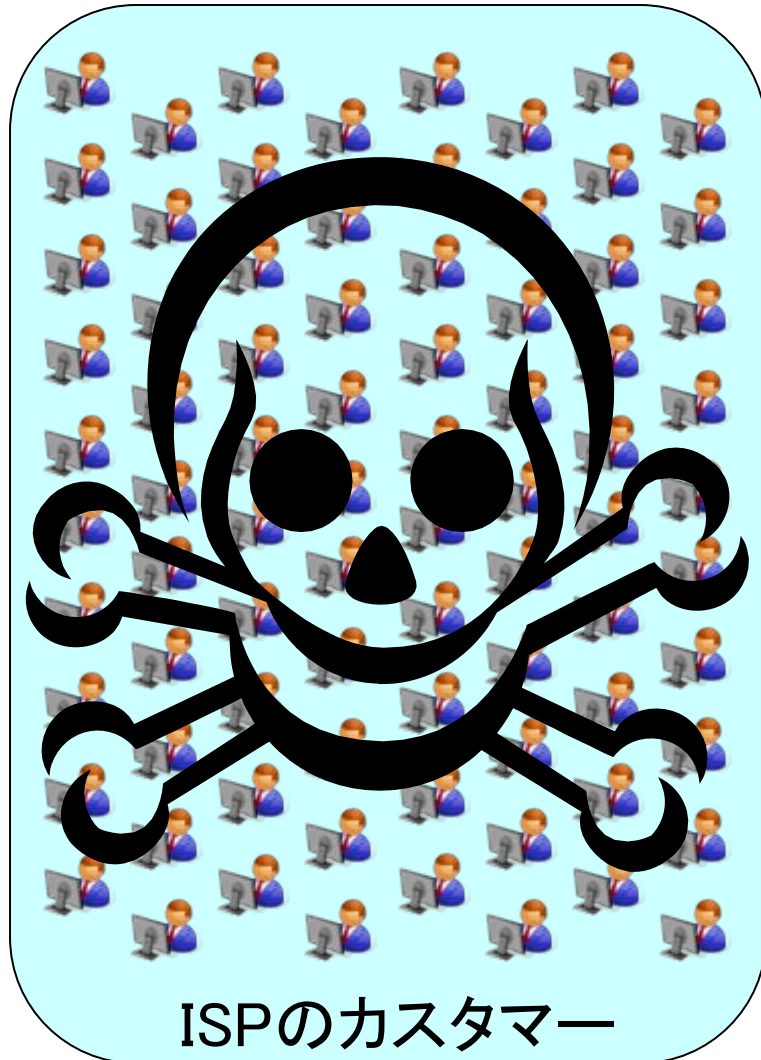
DNSの正常な流れ(2回目以降)



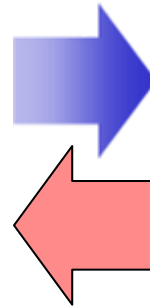
DNSへの毒入れ攻撃



ISPのキャッシュDNSサーバが 狙われたら



ISPのキャッシュ
DNSサーバ



顧客全員が
被害に会う

毒入れ攻撃

- ユーザは正常なアクセスを行っているつもりでも、フィッシングサイトに誘導される
 - 攻撃されたことに気づきにくい
- 同じキャッシュDNSサーバのユーザ全員が影響を受ける
 - 大手ISPのキャッシュDNSサーバが攻撃されると被害は甚大
- 攻撃そのものの検出が容易ではない
 - キャッシュへの毒入れは、見た目は通常のDNSパケットであるため、正常な応答と攻撃の区別が簡単ではない
- 画期的な毒入れ手法(攻撃成功率ほぼ**100%**)である、Kaminsky型攻撃手法の公開(2008年7月)

毒入れへの対策

- Kaminsky型攻撃手法への対策
 - 攻撃成功確率を下げるパッチや、その手法を取り込んだ実装の採用
 - ⇒ 対症療法であり、執拗な攻撃には無力
- 毒入れへの根本対策
 - DNSプロトコルそのものが持つせい弱性であり、完全対処にはDNSのセキュリティ面でのプロトコル拡張が必要
 - ⇒ このための技術が**DNSSEC**

DNSSECとは

- DNSセキュリティ拡張
(DNS SECurity Extensions)
 - 公開鍵暗号を使い、検索側が受け取ったDNSレコードの**出自・完全性**(改ざんのないこと)を検証できる仕組み
 - 従来のDNSとの**互換性を維持した拡張**
- キャッシュへの毒入れを防ぐことができる、現時点で唯一の現実解
 - 他の技術も存在するが標準化が成されていない

世界のDNSSEC情勢

- 導入済(試験的導入も含む)

ccTLD **.se** .pr .bg .br .cz .th .tm .uk

gTLD .museum .gov .org

- 導入予定

ccTLD .ca .ch .cn .de .gr **.jp** .kr .li .my .ru

.jpは2010年中を目処に導入予定

gTLD .biz .cat **.com** .edu .info **.net**

.com / .netは2011年前半に導入予定

DNSSECの導入は、世界の流れ

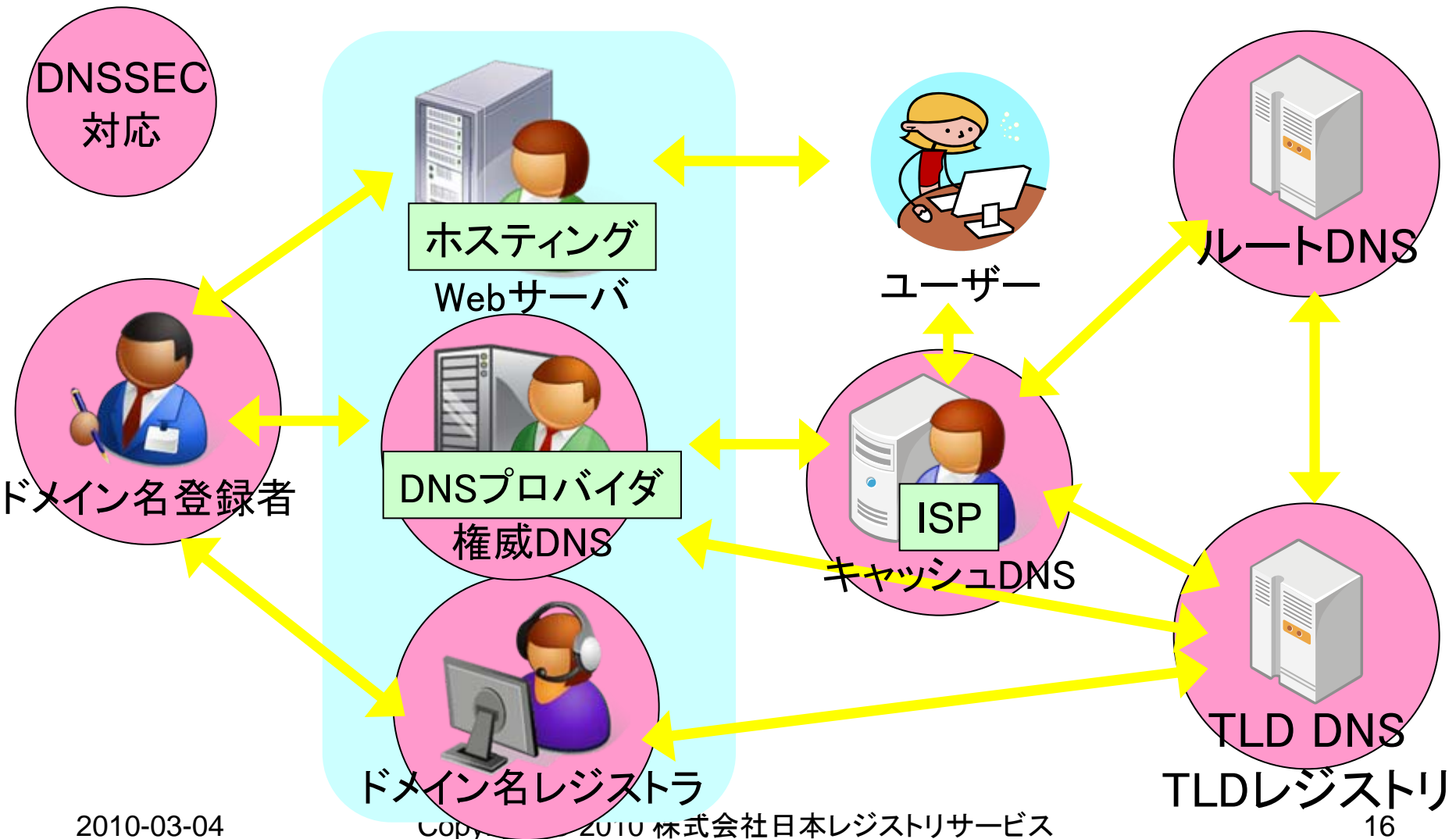
rootゾーンの状況

<http://www.root-dnssec.org/>

- 2010年1月より各rootサーバに**徐々に適応**し、2010年7月から正式運用
 - DNSSEC導入によるDNSデータの変化に対するインパクトが不明確であるため慎重に対応
- 検証できないダミーの署名データを追加したrootゾーン(DURZ)を使い、各rootサーバに順に適応
 - L ⇒ A ⇒ M, I ⇒ D, K, E ⇒ B, H, C, G, F ⇒ J
2010年7月の正式導入までに完了予定
 - 現時点でL, A, M, IにDURZを導入済み
⇒ 問題は発生していない

ホスティングでの DNSSEC対応作業

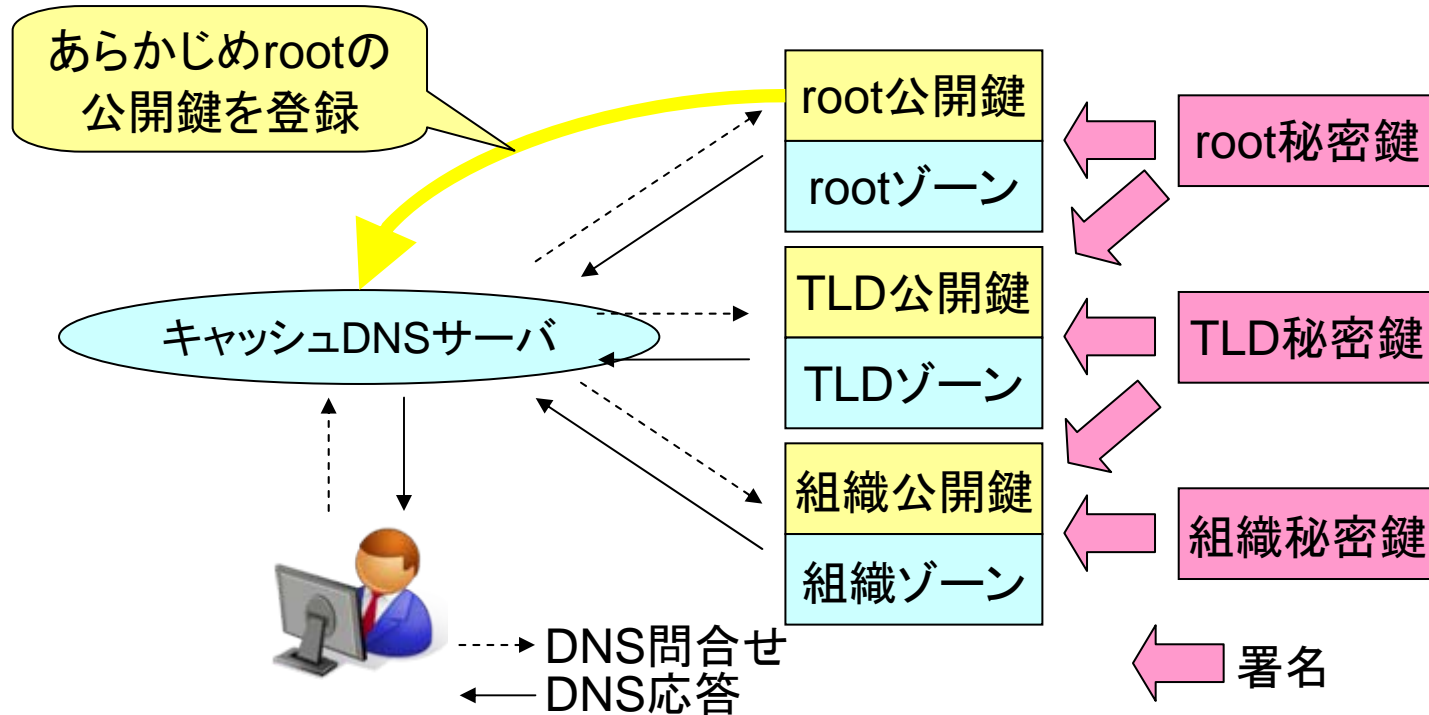
DNSSEC対応が必要な関係者



DNSSEC対応作業の概要

- ドメイン名登録者
 - DNSSEC導入の決定
- ドメイン名レジストラ
 - 鍵の上位への取次ぎ
- TLD DNS、ルートDNS
 - 権威DNSサーバのDNSSEC対応化
 - ゾーンへの署名
- DNSプロバイダ
 - 権威DNSサーバのDNSSEC対応化
 - 秘密鍵・公開鍵を作成し、
てゾーンに署名
- ISP
 - キャッシュDNSサーバのDNSSEC対応化
 - (キャッシュDNSサーバでの)署名の検証

DNSSECの信頼の連鎖の概念図



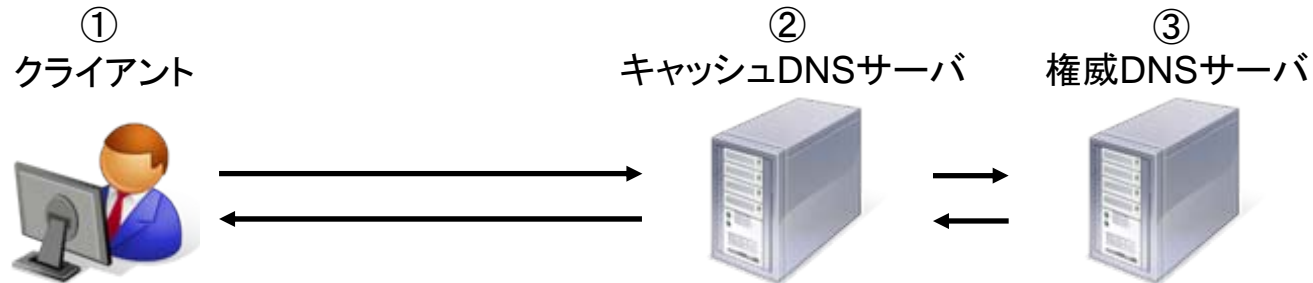
- 秘密鍵で、自ゾーンと下位ゾーンの公開鍵に署名
- rootの公開鍵をキャッシュDNSサーバ(バリデータ)に登録することで、rootから組織までのゾーンの信頼の連鎖を確立

用語:バリデータ(Validator)

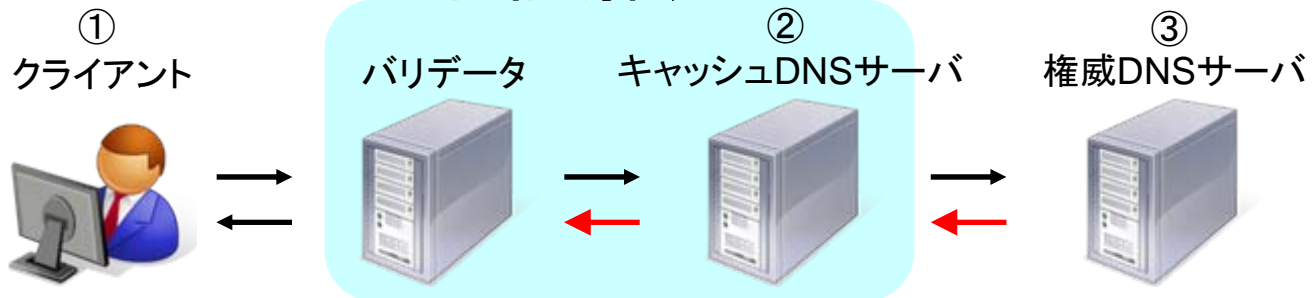
- DNSSECにおいて、バリデータは署名の検証を行うもの(プログラム、ライブラリ)を指す
⇒DNSSECは、**権威DNSサーバからバリデータまでのDNSデータを保証**
- バリデータの所在
 - キャッシュDNSサーバが署名検証を行う場合、キャッシュDNSサーバがバリデータ
⇒ 現状、もっとも一般的なDNSSECのモデル
 - WEBブラウザ等のDNS検索を行うアプリケーションが直接署名検証を行うモデルも考えられる

DNSSEC化による 名前解決モデルの変化

- 従来のDNSでの名前解決モデル



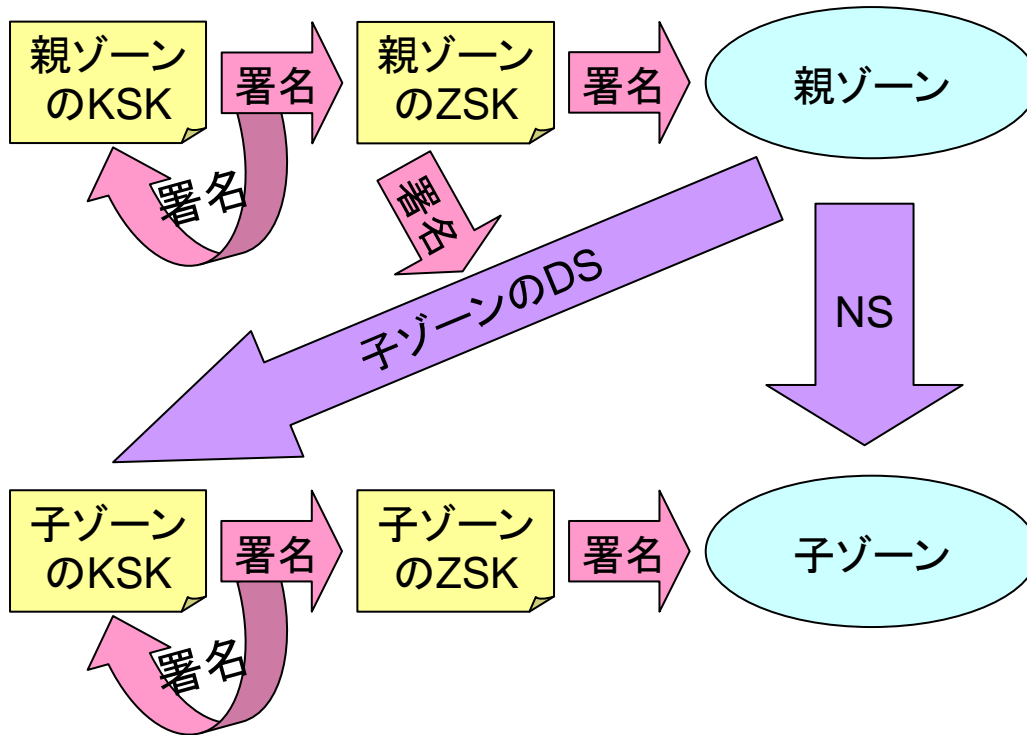
- DNSSECでの名前解決モデル



– 現状バリデータは②に実装

– バリデータが①に実装されているモデルもあり得る

DNSSECの信頼の連鎖の詳細



- **KSK:** 鍵署名鍵
 - 鍵情報に署名し、ゾーンに登録
 - KSKの情報を**DS**として上位ゾーンに登録
- **ZSK:** ゾーン署名鍵
 - ゾーン情報に署名
- 2種類の鍵を使って信頼の連鎖を形成
- 上位のKSK公開鍵をバリデータに登録し、署名検証

ゾーン情報のDNSSEC化

- 公開鍵・秘密鍵の生成
 - KSK: 暗号強度の高い鍵(例: RSA2048bit)
⇒ DS登録の関係で比較的長期間利用
 - ZSK: 暗号強度の低い鍵(例: RSA1024bit)
⇒ ゾーン署名時の負荷を考慮
- 生成した鍵でゾーン情報へ署名
- 上位ドメインへDS情報の登録
 - DS: ハッシュ関数(SHA-1、SHA-256等)によってKSK公開鍵を圧縮したもの

署名前のゾーンファイルの例

```
$TTL      1D
$INCLUDE  example.jp.keys
@         IN          SOA      ns root (
                        1       ; Serial
                        10800    ; Refresh
                        3600     ; Retry
                        3600000  ; Expire
                        1800    ) ; Minimum TTL

                        NS       ns
                        MX       10 mail

;
ns        A          192.0.2.17
www       A          192.0.2.18
mail     A          192.0.2.19

sub1      NS         ns.sub1
ns.sub1  A          192.0.2.49

sec3      NS         ns.sec3
ns.sec3  A          192.0.2.65
$INCLUDE  ../sec3.example.jp/dsset-sec3.example.jp.

sub3      NS         ns.sub3
ns.sub3  A          192.0.2.81
```

署名済みゾーンファイルの例(抜粋)

```
; File written on Tue Nov 10 16:48:50 2009
; dnssec_signzone version 9.7.0b2
example.jp.          86400   IN  SOA  ns.example.jp. root.example.jp. (
                    1257839330 ; serial
                    <中略>
                    )
                    86400   RRSIG  SOA 7 2 86400 20091210064850 (
                    20091110064850 23522 example.jp.
                    CDq8qzNsLVa6pRD9VUE71IYzIaO7u5NtYwwM
                    <中略>
                    UMHqKQinfJHi/8hv4ff5FK198Dc= )
                    86400   NS      ns.example.jp.
                    86400   RRSIG  NS 7 2 86400 20091210064850 (
                    20091110064850 23522 example.jp.
                    UICLoNT5Zszv8LzF0mrkslDMwf9KBmiRSbhN
                    <中略>
                    oY1VNG0n6B+Q2ksY12ZXLK4G0yw= )
                    86400   MX      10 mail.example.jp.
                    86400   RRSIG  MX 7 2 86400 20091210064850 (
                    20091110064850 23522 example.jp.
                    TQz52cCZQvpgcMFyRPtM2BWKxE8Vfvj/RmSv
                    <中略>
                    7GKlXyx3aHYyX3w9O03iXFQz7PA= )
                    86400   DNSKEY  256 3 7 (
                    AwEAAfzJXPiYtSD8DJs+J36dZd+cNrXHxLpu
                    <中略>
                    Zl0VvPOGMNC94WFM+RciLySk2QSoJz mz
                    ) ; key id = 23522
                    86400   DNSKEY  257 3 7 (
                    AwEAAe1MfTlcaIiidHDoCmmhuizPPoO5Tzzh
                    <中略>
                    wZmOr6UvsYzCJLLJsYb9HH8=
                    ) ; key id = 21891
```

鍵更新

- 同じ鍵を使い続けるのはリスクがあるため、定期的に鍵を更新する
 - KSKは比較的長期間(例:1年)で鍵更新
⇒ 上位ゾーンのDS情報の更新が必要
 - ZSKは比較的短期間(例:1~3ヶ月)で鍵更新
⇒ 自ゾーンのみで作業が完結
- DSや、KSK、ZSKの公開鍵の情報はDNSデータであるためキャッシュ対象
 - キャッシュ時間(TTL)を考慮して作業する

ゾーン情報の再署名

- 署名には有効期限があり、長すぎるのは不適切
 - 万が一の事態(鍵の盗難等)において速やかに対応するためには、署名期間は短いほうがよい
- 有効期限が数分の鍵も技術的には可能
 - 休日等の対応を考慮すると現実性に欠けるため、適切な期間を設定する(例:3日~2週間程度)
- 署名の有効期限に達する前に、署名の有効期限を更新するため**ゾーン全体の再署名**が必要となる
 - DNSSEC運用では、ゾーン内の登録レコードに変化が無くても**定期的な署名作業**が必要

DNSSEC化によるDNSデータの変化

- 署名の付加によりゾーンデータが大きくなる
 - 5～10倍程度(鍵のbit長に依存)
 - プライマリが署名すると、セカンダリにもインパクトがある
- DNS応答パケットのサイズが大きくなる
 - DNSトラフィックが増える
 - キャッシュDNSサーバ側では、同じメモリでキャッシュできるレコード数が減る
 - ⇒ キャッシュ効率が落ちる

鍵と署名に関する計算負荷

- 負荷の比較： 検証 << 署名 << 鍵生成
 - 計算負荷は鍵のビット長が長い程高くなる
- 複数ドメインを管理する
 - ⇒ ドメイン名数分の作業を繰り返す
 - 例) 1万ドメイン名をDNSSEC化
 - ドメイン名数分の鍵生成(KSKとZSK)
 - 1万個のゾーンに対して定期的な署名
 - 鍵更新時期には、新たな鍵の生成作業
 - 例えば、ZSKは1ヶ月毎、KSKは1年毎に鍵生成

DNSSEC導入によるリスク

- 署名検証に**失敗した**場合、名前解決**不能**
⇒ 目的のサーバにアクセスできなくなる
- 署名検証が失敗する主な要因
 - 鍵を取り違えた
 - 署名に使う鍵
 - 上位に登録する鍵
 - バリデータに登録する鍵
 - 署名の有効期間を過ぎた
⇒ DNSSECではサーバの時刻が意味を持つ

DNSSECでは、より慎重な運用が求められる

ホスティングでのDNSSEC対応作業 まとめ

- 従来のDNS(DNSSEC無し)との比較
 - KSKとZSKを作成し管理する必要がある
 - 定期的にゾーンに署名を行う必要がある
 - 定期的に鍵更新を行う必要がある
 - 子ゾーンではKSKを更新する度に親ゾーンにDSの登録作業を行う必要がある
- 鍵管理と、ゾーン署名のコストが増大する
⇒ DNSに関する運用コストの増大

DNSSECジャパン (DNSSEC.jp)

- DNSSECの普及を目的として、
多くの関係者が集う場 (現在22会員)
<http://dnssec.jp/>
- DNSSECの導入・運用に関する
 - 課題の整理・共有
 - 技術検証の実施、ノウハウの蓄積
 - BCPの策定
- 成果の対外的発信によるDNSSECの普及・啓発

Q and A

