# Domain Name System (DNS) e Security Extensions (DNSSEC): alcuni aspetti di Privacy e Sicurezza - Parte 2

**Author :** Andrea Pasquinucci

**Date :** 20 Maggio 2019


Nella prima parte di questo articolo è stato descritto brevemente il protocollo DNS e sono state indicate le principali debolezze di sicurezza che lo affliggono. In questa seconda parte dell'articolo presenteremo il protocollo DNSSEC, che si propone di risolvere buona parte di queste debolezze.

## DNS Security Extensions (DNSSEC)

Il processo di studio, implementazione e valutazione di una nuova versione del protocollo DNS che garantisca autenticità e integrità dei dati è incominciato almeno 20 anni fa e non si è ancora concluso. Già solo questo intervallo temporale indica la difficoltà di introdurre delle nuove misure di sicurezza nel protocollo senza stravolgerlo e garantendo l'interoperabilità con quanto in essere.

Vi sono state molte proposte diverse, ma quella ad uno stadio avanzato di implementazione si chiama DNSSEC. L'idea di base è molto semplice: firmare digitalmente tutti i record DNS. L'implementazione si è mostrata invece molto ardua. La seguente è una descrizione a livello molto alto del protocollo (per i dettagli, si vedano tra gli altri gli RFC 4033, 4034, 4035).

Scopo principale di DNSSEC è quello di apporre una firma digitale all'insieme di dati detto Resource Record Set (RRSet) che forniscono la risoluzione di nome nei relativi indirizzi IP con i parametri associati. Quindi nel server DNS d'autorità per un certo nome a dominio, oltre ai dati DNS, ovvero il RRSet, sono presenti anche un record RRSIG che contiene la firma digitale sul RRSet, e un record di tipo DNSKEY che contiene la chiave pubblica necessaria per la verifica della firma digitale RRSIG e corrispondente alla chiave privata utilizzata per creare la firma digitale. Queste coppie di chiavi pubbliche-private sono chiamate **Zone Signing Key** pair (ZSK). Si noti che una DNSKEY che contiene la chiave pubblica di una ZSK è identificata dal flag 256.

Ovviamente la firma digitale è effettuata in un ambiente sicuro, tipicamente off-line e la chiave

privata di firma è, ove possibile, tenuta in un modulo hardware apposito.

Con la firma digitale sui RRSet viene assicurata l'integrità dei dati, ma per il momento non è risolto il problema dell'autenticità perché un attaccante potrebbe sostituire la tripletta RRSet, RRSIG(RRSet) e DNSKEY(ZSK) con i propri dati, ad esempio tramite attacchi di tipo “DNS spoofing” e “cache poisoning” descritti precedentemente.

Il problema è, quindi, garantire l'**autenticità delle chiavi ZSK**, ovvero che queste siano proprio quelle create e gestite dal titolare del dominio DNS sul server DNS d'autorità.

Si potrebbe pensare di introdurre un repository centrale per queste chiavi alla PGP, oppure adottare l'approccio delle Certification Authorities come per i certificati utilizzati per i siti web. Questi e altri approcci simili, però, avrebbero notevoli conseguenze sul protocollo DNS originario e non sarebbero facilmente integrabili in esso.

DNSSEC, invece, segue l'approccio del protocollo DNS originario che, come descritto precedentemente, è basato sulla presenza dei 13 root server di cui sono noti pubblicamente a priori gli indirizzi IP, e della catena di fiducia (trust) a partire da questi. Vi sono quindi due punti da risolvere: come gestire l'origine dell'albero delle firme digitali e come passare da un ramo al successivo mantenendo la fiducia. DNSSEC risolve questi punti introducendo un nuovo tipo di chiavi[1] chiamate **Key Signing Key** (KSK), la cui parte pubblica è disponibile sempre in record di tipo DNSKEY. Si noti che le KSK sono chiavi “self-signed” e che una DNSKEY che contiene la chiave pubblica di una KSK è identificata dal flag 257.

Tutto l'albero di fiducia è attestato su di una chiave, la KSK(ROOT), si veda la Fig. 5.

```
; // The root key in bind format. This can be read by most tools, including
; // named, unbound, et. For libunbound, use ub_ctx_trustedkeys() to load this
trusted-keys {
"." 257 3 8
"AwEAAaz/tAm8yTn4Mfeh5eyI96WSVexTBAvkMgJzkKTOiW1vkIbzxeF3+/4RgWOq7HrxRixH
lFlExOLAJr5emLvN7SWXgnLh4+B5xQlNVz8Og8kvArMtNROxVQuCaSnIDdD5LKyWbRd2n9
WGe2R8PzgCmr3EgVLrjyBxWezF0jLHwVN8efS3rCj/EWgvIWgb9tarpVUDK/
b58Da+sqqls3eNbuv7pr+eoZG+SrDK6nWeL3c6H5Apxz7LjVc1uTIdsIXxuOLYA4/
ilBmSVIzuDWfdRUfhHdY6+cn8HFRm+2hM8AnXGXws9555KrUB5qihylGa8subX2Nn6UwNR
1AkUTV74bU=" ; // key id = 20326

"." 257 3 8
"AwEAAagAIKlVZrpC6Ia7gEzahOR+9W29euxhJhVVLOyQbSEW0O8gcCjFFVQUTf6v58fLjw
Bd0YI0EzrAcQqBGCzh/
RStIoO8g0NfnfL2MTJRkxoXbfDaUeVPQuYEhg37NZWAJQ9VnMVDxP/VHL496M/QZxkjf5/
Efucp2gaDX6RS6CXpoY68LsvPVjR0ZSwzz1apAzvN9dlzEheX7ICJBBtuA6G3LQpzW5hOA2hz
CTMjJPJ8LbqF6dsV6DoBQzgul0sGIcGOYl7OyQdXfZ57relSQageu+ipAdTTJ25AsRTAoub8ON
GcLmqrAmRLKBP1dfwhYB4N7knNnulqQxA+Uk1ihz0="; // key id = 19036
};
```

Fig. 5 La chiave pubblica KSK(ROOT) all'origine dell'albero di autenticità DNSSEC

In realtà in Fig. 5 sono presenti due chiavi ma, come indicato in https://data.iana.org/root-anchors/root-anchors.xml, la chiave con id=19036 è scaduta il 2019-01-11 e rimane solo per retro compatibilità.

Questo ci porta ad affrontare le modalità di gestione delle chiavi KSK(ROOT). Vista l'enorme importanza di questa chiave, la procedura di creazione e la sua gestione sono soggette a misure di sicurezza estremamente complesse ed estese. Ad esempio la procedura di creazione della chiave KSK(ROOT) include una cerimonia con la presenza fisica di autorità delegate a questo e la produzione di evidenze conservate fisicamente in luoghi separati, si veda https://www.iana.org/dnssec/files per ulteriori informazioni. La chiave KSK(ROOT) è usata raramente e la procedura di sostituzione è lunga e complessa, anche perché questa chiave deve essere distribuita manualmente a tutti i server e resolver DNS, insieme agli indirizzi IP dei 13 root server. Inoltre si deve tenere conto dei TTL che indicano i tempi di permanenza dei dati nelle cache DNS in tutto il mondo. Pertanto la chiave KSK(ROOT) deve durare a lungo: la prima chiave è stata introdotta nel 2010 e, come visto, sostituita solo all'inizio del 2019 dalla seconda chiave che è stata generata nel 2016 (si veda ancora https://www.iana.org/dnssec/files).

La chiave KSK(ROOT) viene utilizzata solo per firmare le chiavi ZSK dei root server, generate e sostituite più frequentemente e utilizzate per firmare gli RRSet presenti nei 13 (identici) root server. Per verificare l'autenticità delle ZSK, bisogna scaricare da un root server la ZSK e la RRSIG(ZSK) - ovvero la chiave pubblica ZSK e la sua firma digitale RRSIG - e verificare la firma digitale RRSIG con la chiave pubblica KSK(ROOT) ottenuta direttamente da fonte fidata.

Ogni autorità DNS genera le proprie chiavi KSK con le quali firma unicamente le proprie chiavi ZKS. Inoltre ogni autorità DNS genera un figerprint (o “hash”) della propria chiave pubblica KSK e la invia in un record chiamato Delegation Signer (DS) alla zona genitore (ovvero il ramo dell'albero superiore) in modo che questa sia associata agli indirizzi IP e agli altri dati dei propri server DNS, e ovviamente sia firmata con la ZSK della zona genitore.

La procedura è sicuramente complessa e rispetto al protocollo originale DNS richiede molti più passaggi, molti più dati e molte più risorse computazionali. Ma DNSSEC è in grado di garantire sia l'integrità che l'autenticità della risoluzione DNS a un costo, ad oggi, sicuramente affrontabile.

Per descrivere ad alto livello come tutto questo funzioni, utilizziamo come esempio di risoluzione il nome a dominio di prova DNSSEC sigok.verteiltesysteme.net:

1. il punto di partenza del nostro resolver DNSSEC è la conoscenza dei 13 indirizzi IP dei root server e della chiave pubblica KSK(ROOT);
2. la prima richiesta a un root server è di fornire la ZSK e la sua RRSIG;
3. utilizzando KSK(ROOT) il resolver verifica la firma RRSIG sulla ZSK; ora la ZSK è integra e autentica;

4. il resolver chiede al server root di risolvere sigok.verteiltesysteme.net, il server root risponde di non conoscere la risoluzione ma invia i dati del server DNS della zona .net in un RRSet firmato con la ZSK;
5. il resolver verifica la firma della ZSK sul RRSet che quindi è integro e autentico; si noti che all'interno del RRSet c'è la fingerprint della KSK della zona .net (record DS);
6. il resolver si collega al server DNS della zona .net indicato dal server root e richiede la chiave pubblica KSK della zona, la chiave ZSK e la firma RRSIG;
7. il resolver verifica che la fingerprint della chiave KSK sia identica a quella presente nel RRSet ricevuto dal root server; ora la chiave KSK della zona .net è integra e autentica;
8. Con la chiave KSK della zona .net, il revolver verifica la firma RRSIG della chiave ZSK; ora la ZSK della zona .net è integra e autentica;
9. il processo continua ripetendo i passaggi da 4 a 8 con i server DNS delle zone delegate scendendo nell'albero DNS sino a che un server risponde inviando un RRSet contente gli indirizzi IP e gli altri dati di sigok.verteiltesysteme.net; a questo punto il resolver verifica con la chiave pubblica dell'ultima ZSK la firma su questo RRSet e si conclude la risoluzione DNSSEC.

Alcuni tool online, quale ad esempio https://www.dnslookup.org/, permettono di visualizzare i principali passaggi di questa procedura e in particolare la verifica delle firme digitali sugli RRSet e  sulle DNSKEY.

Un'ultima osservazione sulle chiavi KSK e ZSK: a differenza degli altri dati delle zone la cui validità temporale è soggetta a un TTL ovvero un intervallo di tempo relativo, queste sono valide sino a una data assoluta. Quindi ogni server e resolver DNS deve avere un orologio allineato all'ora esatta. Prima di DNSSEC questo non era un requisito essenziale: ora lo diviene e apre la possibilità, anche se remota, di attacchi basati sulla retro-datazione dell'ora nei server DNS.

## DNSSEC e la cache

Come indicato precedentemente, l'efficienza della risoluzione DNS è basata sull'uso delle cache e dei **resolver ricorsivi**. Anche i dati DNSSEC possono essere mantenuti in cache secondo le logiche dei TTL. I resolver ricorsivi DNSSEC inviano le loro richieste con il flag “DNSSEC OK” (DO) attivo, il che richiede che il server DNS risponda con tutti i dati DNSSEC, o informando il resolver che DNSSEC non è disponibile per le informazioni richieste. Si noti che il flag “DO” fa parte delle estensioni EDNS, in ogni caso necessarie per poter gestire tutti i dati DNSSEC.

Ovviamente la cache di un resolver ricorsivo DNSSEC è molto più grande della corrispondente cache DNS originale, in quanto devono essere gestite anche tutte le firme e le chiavi pubbliche necessarie per la loro verifica.

Si noti anche la completa interoperabilità di DNSSEC con DNS, in quanto se un server DNS risponde negativamente alla richiesta del flag “DO”, la risoluzione procede esattamente come prima. Altrimenti i dati DNSSEC sono dati in più rispetto al protocollo originale.

Più complessa è la situazione degli ***stub resolver***, ovvero dei resolver presenti sui dispositivi

client. Uno stub resolver potrebbe verificare indipendentemente tutte le firme digitali e tutti i dati che riceve, ma questo imporrebbe sia un elevato carico di lavoro sui client che un enorme carico sui server DNS, particolare sui server root e TLD. La configurazione tipica di un stub resolver è invece quella di essere “non validating”. In questo caso lo stub resolver si fida della risoluzione e delle verifiche fatte dal resolver ricorsivo a cui si connette. Se il resolver ricorsivo risponde con il flag Authenticated Data (AD) vuol dire che DNSSEC è valido e la risoluzione è integra e autentica, si veda la Fig. 6.

```
$ dig +dnssec +multi sigok.verteiltesysteme.net
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51654
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;sigok.verteiltesysteme.net. IN A

;; ANSWER SECTION:
sigok.verteiltesysteme.net. 60 IN A 134.91.78.139
sigok.verteiltesysteme.net. 60 IN RRSIG     A 5 3 60 (
                                20190730020006 20190430020006 30665 verteiltesysteme.net.
                                LrXt0utcEKK/D/sbscWNAoAT7kLYt9zUEJKxC4rFH3ZE
                                69VUjLmCTEhBv1Xoo5QQUv3NkZeVBbs8VcRaGX4wmHiN
                                NNu5PrP3SBILmICuT/pkM8n8Ex5PH3KKnNsaLwZmwo9V
                                0ziZsu6byNyjyV0Cj05hDzcXv4d1zHPmRNEcG3Y= )
```

Fig. 6 Esempio di risoluzione DNSSEC valida

Se invece il processo di verifica delle firme digitali fallisce, la risoluzione non va a buon fine (l'indirizzo IP, in questo caso 134.91.78.139, non viene fornito anche se presente nei dati ricevuti dal server DNS) e viene riportato un messaggio di errore con codice SERVFAIL e senza il flag AD, come in Fig. 7.

```
$ dig +dnssec +multi sigfail.verteiltesysteme.net
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 31280
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;sigfail.verteiltesysteme.net. IN A
```

Fig. 7 Esempio di risoluzione DNSSEC con errore di verifica delle firme digitali

Nel caso invece la zona DNS non sia gestita con DNSSEC, il processo di validazione DNSSEC si interrompe quando non viene trovato un record DS che permette di scendere nell'albero di risoluzione e si prosegue con il processo DNS originario come in Fig. 8.

```
$ dig +dnssec +multi www.wikipedia.org
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 28134
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;www.wikipedia.org.  IN A

;; ANSWER SECTION:
www.wikipedia.org.  106 IN A 91.198.174.192
```

Fig. 8 risoluzione DNS in zone senza DNSSEC

Infine, un **problema** di non semplice soluzione riguarda la dimostrabilità dell'assenza di un nome a dominio. In questo caso il protocollo DNS originale fornisce una risposta vuota, ma questa è facilmente gestibile da parte di un attaccante che può rimuovere tutti i dati presenti in cache od in transito verso un resolver. DNSSEC fornisce invece una risposta autentica che dimostra l'assenza di un nome a dominio utilizzando i record detti Next Secure Record (NSEC o NSEC3). NSEC3 fornisce la prova che, nella lista ordinata di nomi del dominio, non vi sia quello richiesto, utilizzando tecniche crittografiche di Hash in modo da non permettere la numerazione dei domini presenti. La risposta DNSSEC alla richiesta di risoluzione di un nome non esistente genera un errore NXDOMAIN e riporta i record NSEC3 firmati, come in Fig. 9.

```
$ dig +dnssec +multi wikapedikkk.org
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 63445
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 8, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;wikapedikkk.org.      IN A

;; AUTHORITY SECTION:
org.                    900 IN SOA a0.org.afilias-nst.info. noc.afilias-nst.info. (
                                2013453931 ; serial
                                1800       ; refresh (30 minutes)
                                900        ; retry (15 minutes)
                                604800     ; expire (1 week)
                                86400      ; minimum (1 day)
                                )
org.                    900 IN RRSIG SOA 7 1 900 (
                                20190523163642 20190502153642 16454 org.
                                aVTPvveCU9bd4nMmaOLe1kWIYQ+C1xJ2V8FQ2BdQFxcT
                                AX2OabGxgwlJJXKS4dR6YMkiLechbe9+nnI9D0GKNlWB
                                SjtQFaQ+psLtiu6wsj3JFNrsmhnlgzmc53jIFUuyTthW
                                VeUrrkfnv6GAxkcKLMMjys8qbzMNMFCvgzfAI7o= )
h9p7u7tr2u91d0v0ljs9l1gidnp90u3h.org. 86400 IN NSEC3 1 1 1 D399EAAB (
                                H9PARR669T6U8O1GSG9E1LMITK4DEM0T
                                NS SOA RRSIG DNSKEY NSEC3PARAM )
h9p7u7tr2u91d0v0ljs9l1gidnp90u3h.org. 86400 IN RRSIG NSEC3 7 2 86400 (
                                20190523163642 20190502153642 16454 org.
                                NQ1LLyMkJDNri2wVTPhlCxi9yLKFefF61KJmr65yU3Sk
                                AnMCBcZlVkpYuOm1KjMlLS7EvLUGD6PqjTyJZ5+qx/Mj
                                C6bjRCW7q+hn5D1Gm6jeHiz34xloznvId0VARmXIxGJb
                                FPr9xK4F7urftlaMvjn7+KcdJ2xYCehiYBOHkJY= )
[...]
```

Fig. 9 esempio di risposta NXDOMAIN NSEC3 alla richiesta di risoluzione di un dominio inesistente in DNSSEC

## Usare oggi DNSSEC

L'utilizzo di DNSSEC è ancora molto limitato. Solamente i livelli superiori dell'albero DNS hanno adottato consistentemente DNSSEC, mentre l'adozione per i nomi completi che usiamo tutti i giorni è ancora molto "a macchia di leopardo". Ad esempio registro.it riporta che ad aprile 2019, su un totale di 1.225 Registrar accreditati per registrare nomi a dominio nella zona .it, solo 19

permettono di registrare zone con DNSSEC. Inoltre, https://stats.dnssec-tools.org/ mostra che ai primi di maggio 2019 nella zona .it sono presenti solo circa 4.400 nomi a dominio DNSSEC su un totale di circa 3.200.000.

Questo vuol anche dire che molti dei resolver DNS ricorsivi che utilizziamo ogni giorno non hanno ancora implementato DNSSEC.

In realtà non è molto complesso adottare un resolver locale DNSSEC. I punti principali da considerare sono tre:

1. identificare un resolver ricorsivo DNSSEC di fiducia soprattutto per quanto riguarda gli aspetti di Privacy discussi precedentemente;
2. configurare una connessione sicura tra il proprio stub resolver ed il resolver ricorsivo DNSSEC (questo sia per motivi di privacy che per impedire attacchi ai dati trasmessi visto che lo stub resolver tipicamente è “non validating”);
3. utilizzare localmente uno stub resolver DNSSEC.

Per la sicurezza della connessione dello stub resolver al resolver ricorsivo è stato introdotto il protocollo DNS-over-TLS (RFC 7858) che permette di trasportare il protocollo DNS all'interno di un canale cifrato con TLS sulla porta TCP/853. Oppure è possibile adottare “DNS queries over HTTPS” (DoH, RFC 8484) a cui si è già accennato precedentemente, o infine DNSCrypt (https://dnscrypt.info/), una soluzione alternativa che permette di stabilire una connessione cifrata anche sulla porta TCP/443 tra uno stub resolver ed un resolver ricorsivo che supporti questo protocollo.

Come esempio, su un client linux è possibile implementare una soluzione locale di stub resolver DNSSEC installando il server DNS unbound (https://www.unbound.net/) con DNS-over-TLS e adottando una configurazione simile a quella brevemente descritta in Fig. 10.

```
# /etc/resolv.conf
nameserver 127.0.0.1

# /etc/unbound/unbound.conf
server:
  [...]
  tcp-upstream: yes
  edns-tcp-keepalive: yes
  # wget https://www.internic.net/domain/named.cache
  root-hints: "/etc/unbound/named.cache"
  # If you want to perform DNSSEC validation, run
  # unbound-anchor before you start unbound
  auto-trust-anchor-file: "/var/lib/unbound/root.key"
  trusted-keys-file: /etc/unbound/keys.d/*.key
  tls-upstream: yes
  tls-cert-bundle: "/etc/unbound/ca-bundle.crt"
  [...]
forward-zone:
  name: "."
  forward-addr: 1.1.1.1@853#cloudflare-dns.com
  forward-addr: 1.0.0.1@853#cloudflare-dns.com
  forward-tls-upstream: yes
  forward-first: no
```

Fig. 10 Esempio di configurazione di unbound come stub resolver con i server Cloudflare come resolver ricorsivi DNSSEC

## Sicurezza attuale di DNS e DNSSEC

Come abbiamo visto, il protocollo DNSSEC è sicuramente complesso da adottare e, malgrado sia stato proposto ormai da molti anni, non è ancora diffuso a sufficienza per poter fornire tutte le garanzie di sicurezza di cui avremmo bisogno. Inoltre DNSSEC non fornisce una sicurezza end-to-end, ovvero sino allo stub-resolver locale, né offre misure per garantire la privacy dell'accesso a internet.

Più in generale, la gestione gerarchica del DNS - e in particolare quella di ICANN dei principali TLD - ha da sempre sollevato **critiche,** sia sulla possibilità di censura sia su possibili influenze commerciali a danno della circolazione delle idee e della libertà di accesso alle informazioni e ai

servizi in internet.

A questo proposito si noti che i protocolli DNS e DNSSEC permettono anche la presenza di più alberi paralleli, ovvero di utilizzare più origini fidate contemporaneamente per la risoluzione di nomi in alberi diversi. Questa possibilità è stata sfruttata ad esempio da OpenNIC, Open Root Server Network (ORSN) e New Nations. Questi servizi DNS sono compatibili con quello principale gestito da ICANN ed aggiungono anche la risoluzione di domini in ulteriori TLD quali ad esempio .ko (Kosovo) e .libre.

Un **approccio alternativo** alla risoluzione dei nomi in indirizzi IP è quello utilizzato per primo da Tor con la risoluzione del TLD .onion gestita all'interno dello stesso protocollo di anonimizzazione. Con l'avvento delle monete virtuali, a partire dal bitcoin, e dell'uso della blockchain, altri servizi alternativi di risoluzione dei nomi sono stati introdotti quali Blockchain-DNS, Emercoin, Namecoin eccetera. Questi servizi si basano tipicamente su protocolli decentralizzati quali la blockchain e permettono la risoluzione di un nome in un indirizzo IP tramite un plugin da installare sul browser web che contatta direttamente questi servizi. Sia per Tor che per Blockchain-dns ecc., la risoluzione del nome in indirizzo IP avviene solo utilizzando l'apposita applicazione o plugin e non è un servizio offerto dal sistema operativo e disponibile generalmente a tutte le applicazioni presenti in un dispositivo informatico.

Vi sono **opinioni contrastanti** sul fatto che sia utile e salutare per Internet la presenza concomitante di servizi paralleli - e anche in concorrenza - per la risoluzione dei nomi in indirizzi IP. Da una parte vi sono logiche di libertà di mercato, di libera scelta e anche di alta affidabilità dei servizi, dall'altra una parcellizzazione del servizio può renderlo più fragile, ad esempio rispetto ad attacchi DdoS o di oscuramento di alcune zone, meno omogeneo e di più difficile accesso.

La risoluzione dei nomi in indirizzi IP è una di quelle attività poco conosciute ma senza la quale non esisterebbe Internet. Malgrado la grandissima efficienza del protocollo che è stato in grado di supportare la crescita esponenziale di Internet degli ultimi 20 anni, rimangono serie **problematiche di sicurezza** relative all'autenticità, all'integrità e alla privacy dell'accesso a Internet. La lenta adozione di DNSSEC sicuramente risolve alcune di queste criticità, ma rimane ancora tanto da fare perché la risoluzione dei nomi in indirizzi IP sia un servizio non solo efficiente ma anche “sicuro”.

## Note:

[1] Si noti che gli RFC DNSSEC non richiedono espressamente di utilizzare due chiavi diverse per KSK e ZSK.

Articolo a cura di **Andrea Pasquinucci**