DNSSEC

The good & very bad

bert.hubert@netherlabs.nl
hubert@fox-it.com

FOX-IT
FOR A MORE SECURE SOCIETY

Agenda

- whoami: Fox-IT & PowerDNS
- Brief recap about DNS & DNSSEC
- DNSSEC: The good
- DNSSEC: The bad
- DNSSEC: The harmful parts
- So.. should we be happy?
- Questions

Fox-IT & PowerDNS

- Fox-IT: Security company, ”for a more secure society”
  - Forensics, audits, fighting cybercrime, communications intelligence, high-end security & ”perfect firewalls”
  - 100 hackers, nerds & investigators
  - We like people with interesting projects!
  - **We need you (come to our booth)**
- PowerDNS: My interesting project. Powers around 30%-50% of all domain names around here, around for 12 years, 10 years as open source

DNSSEC & DNS: recap

- DNS gets us to the IP addresses
- If DNS is wrong, our data (and euros!) flow the wrong way
- After https, smtp/tls, pop3s, imap3s, it was felt that DNS should also gain an extra s
- DNSSEC is an attempt to secure DNS, while maintaining all the good bits
  - Redundancy, very high performance, flexibility
- This made it impossible to 'just add an s'

Hoe is DNSSEC anders?

- Om SSL toe te voegen aan email of web is alleen een certificaat benodigd
- Voeg het certificaat toe, herconfigureer, en http://uwbedrijf.nl is ook bereikbaar op https://uwbedrijf.nl
- SSL beveiliging wordt 'live' toegevoegd
    - Vers voor iedere verbinding
- DNSSEC beveiliging wordt doorgaans statisch en **vooraf** toegepast
    - En moet steeds bijgewerkt worden
- Kortom → big deal!

DNSSEC protects against

- ”Spoofing attacks”
  - Large amounts of spoofed packets with 'improved answers' try to get accepted as the real thing
- Unreliable secondaries/slaves
  - Your slave/secondary might fiddle with your data
- Unreliable governments and service providers
  - Might inject advertisements or redirect your vital facebook updates

FOX-IT
FOR A MORE SECURE SOCIETY

DNSSEC: How compelling?

- The threats on the previous page are not immediate
    - Post RFC5452 spoofing attacks are very hard, you can pick your secondaries with care, and governments don't need DNS to get your packets.
- There is no burning need
    - Compare IPv6 which has one
- There are other good reasons though

Security is a feeling

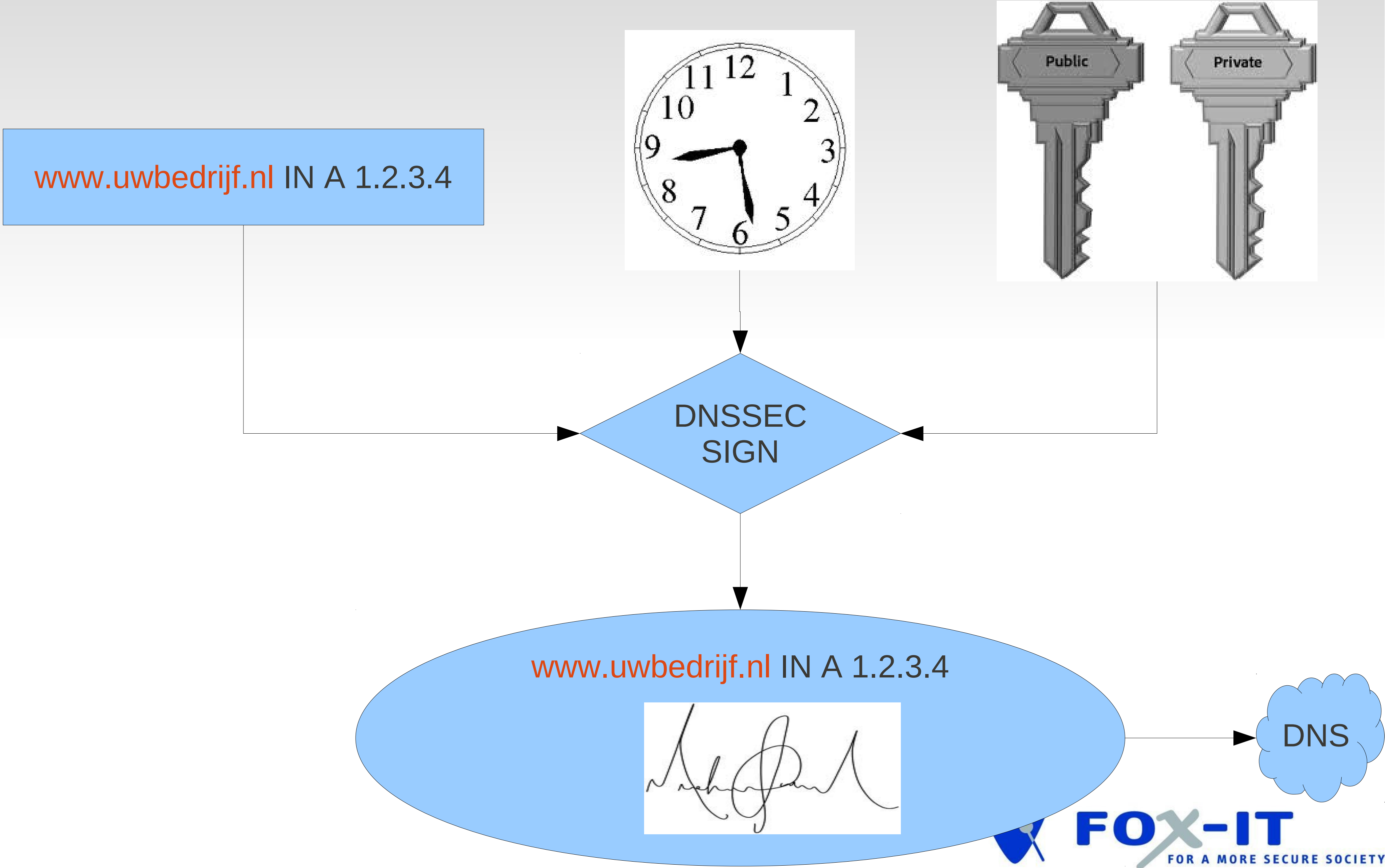
DNSSEC reasons

- Not only ”nice to have”
- Customers start arriving with printouts in hand claiming you don't support it
  - ”Nessus effect”
- Competition is doing it
- Requirement in procurement
- .. great excuse to clean up your DNS!

The very core of DNSSEC

- A DNS response gets a digital signatured called an RRSIG
- This signature is made with a private ejky, the associated public key is published as a DNSKEY record
  - I've always wondered how 'keys' make 'signatures'. Normal people use a pen
- A hash of the DNSKEY (the 'DS') is published in the parent zone, allowing the world to verify your DNSKEY
  - All the way up to the root

The signing process



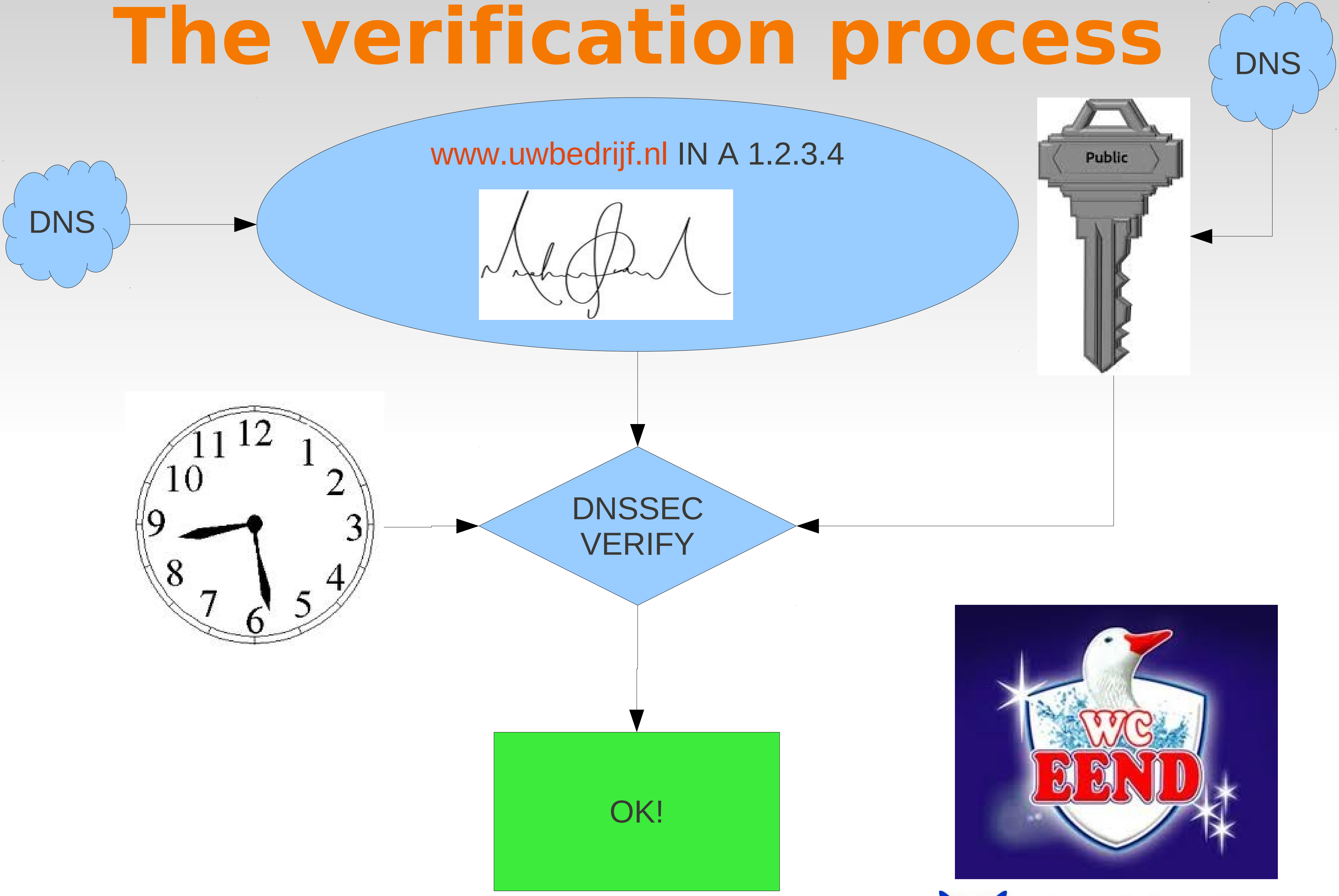
www.uwbedrijf.nl IN A 1.2.3.4

DNSSEC SIGN

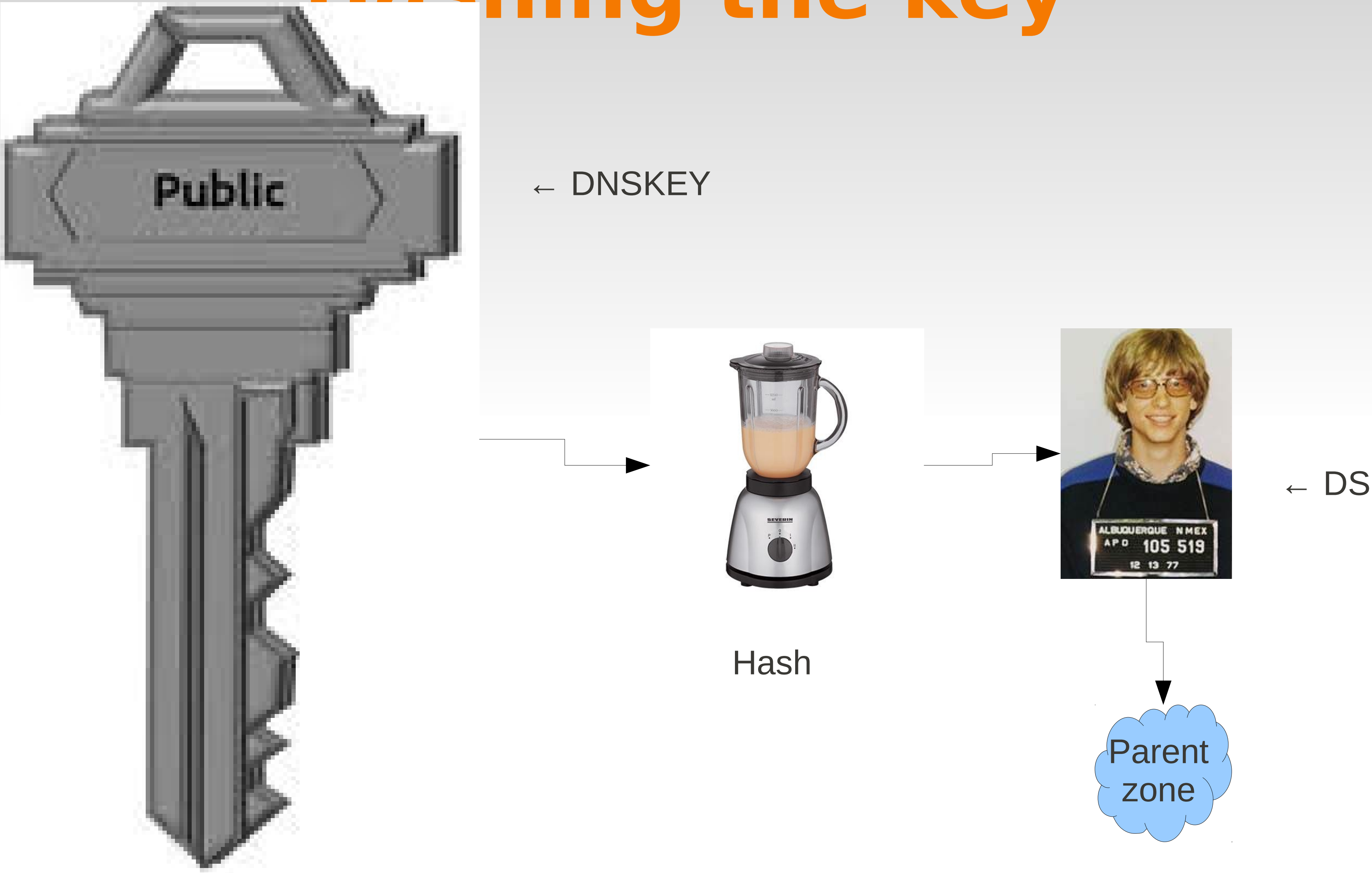
www.uwbedrijf.nl IN A 1.2.3.4

DNS

The verification process



Hashing the key



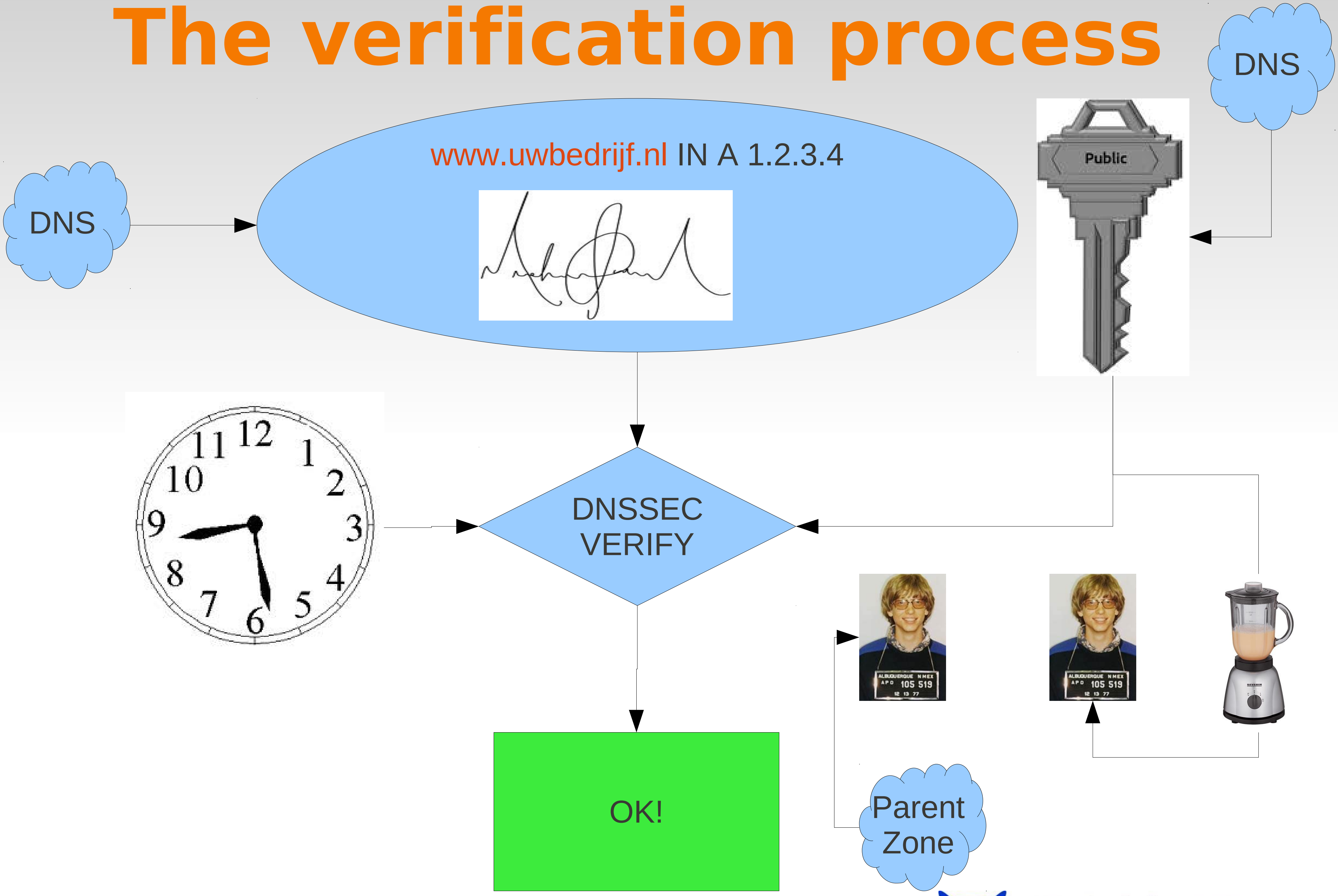
← DNSKEY

Hash

← DS

Parent zone

The verification process



DNSSEC: the good

- It Works <TM> (RFC 1925 compliant)
- We got our own open source open access PKI!!!1!!
  - No pesky CA vendors invoved
  - Very fast
- The cryptographic primitives used are pretty ok, and we can add more (RSA, DSA, SHA256, soon ECDSA, already Russian GOST)
- It is pretty compatible with existing DNS

DNSSEC Design

'There are two ways of constructing a software design:

- One way is to make it so simple that there are obviously no deficiencies,
- and the other way is to make it so complicated that there are no obvious deficiencies.'

C.A.R. Hoare, 1980 Turing Award Lecture

DNSSEC Design

- RFC 4033: DNS Security Introduction and Requirements
- RFC 4034: Resource Records for the DNS Security Extensions, Protocol Modifications for the DNS Security Extensions
- RFC 4035: Protocol Modifications for the DNS Security Extensions
- RFC 4509: Use of SHA-256 in DNSSEC Delegation Signer (DS) Resource Records (RRs)
- RFC 5155: DNS Security (DNSSEC) Hashed Authenticated Denial of Existence
- RFC 5702: Use of SHA-2 Algorithms with RSA in DNSKEY and RRSIG Resource Records for DNSSEC
- → **I guess we went for option 'B'**

DNSSEC design

- DNS is not SQL, not a simple query language
- Answers can be:
  - The real answer
  - A referral to a Canonical Name
  - A delegation to another nameserver
  - A wildcard-synthesized answer
  - ”No such name” (EMPTY!)
  - ”No such record type” (EMPTY!)
- DNSSEC signing must be static!

DNSSEC design: how to sign?

- A direct answer or a CNAME → sign it
- A delegation → oops, nothing to sign!
  - The NS records are owned by the child zone and will be signed with the child key → shit
- Wildcard synthesis → oops, new answer to sign for every query. We sign the wildcard, and provide **each** answer with a flag that says if and how it was generated from that static wildcard → complicated

DNSSEC design: how to sign?

- ”No such name, no such record type” → encoded as an empty answer in DNS. How to sign the void?
    - Generic ticket to denying service if you got it!
- This stumped people for a while
- AH!! If you have hosts called A, B and D, and a question comes in for C, return a special answer that says 'there are no records between B and D' → sign that
- Smart!

DNSSEC Design: how to sign

- Repeating, we have hosts A, B and D.
- To deny existence of E, state that there are no hosts between D and A → sign that
  - Again, smart!
  - A → B NSEC
  - B → D NSEC
  - D → A NSEC
- But wait a moment! This is a virtual listing of all hosts in a zone! We've reinvented the AXFR!

DNSSEC Technique: NSEC Walking

- From this we get a nice trick, virtual zone transfers
- Try: dig -t nsec isc.org
- Gives: isc.org → _kerberos.isc.org
- Try: dig -t nsec _kerberos.isc.org
- Gives: _kerberos.isc.org → _caldav._tcp.isc.org.
- This way we find sql1.isc.org, which is interesting!

DNSSEC Hashed Denial of Existence

- It was long argued that this information leakage was **not a problem**
  - Many begged to differ
  - An impasse was reached
- It took 60 pages of RFC to rectify this situation, and some of the smartest people on the planet. The result is NSEC3
  - And it makes you cry
- It was a giant effort but it appears to work

DNSSEC Hashed Denial of Existence

- When a question comes in for name/record/type that does not exist, calculate the hash of that name
    - Salted, iterated, sha1
- Look al all other names in a zone that have already been hashed in the same way
- Supply an answer that says 'between this hash and that hash, there are no answers'
- But does it help? DJB ploughs through hashes and has reversed loads of names..

On the delegation issue

- Each name in DNSSEC has exactly **ONE** signature(set)
- So if ns1.fox-it.com is part of the .com zone, AND part of the fox-it.com zone, it will only be signed in the fox-it.com zone
  - And not in com!
- So how do we perform a secure delegation?
- **WE DON'T!**
- So if your zone is not signed, but .com is, you don't benefit at all

On the delegation issue

- Seriously?
- So, what IS signed?
- If your zone is not DNSSEC secured, like fox-it.com for example, there will be cryptographic proof of that (!)
  - Thanks dudes
  - Against downgrade attacks!
- If your zone IS signed, verification only really happens at the very end
  - The delegating answer from COM is not verified

DNSSEC technique: denial of service

- Since delegating answers, for example from .com, are not themselves DNSSEC secured, they can be modified at will
- For example, to point at 127.0.0.1
- Since DNSSEC verification only happens at the end
    - Or in this case, not at all
- This means that DNSSEC does nothing to protect the interim resolution steps

Traffic amplification

- The original goal was to keep DNS unmodified, but simply add signatures
- Turned out to be difficult, so changes had to be made (NSEC, NSEC3)
- Part of these changes meant that DNSSEC answers can be HUGE
- dig -t any se +dnssec @a.ns.se → 4KB answer
- Original 'limit' was 512 bytes

DNSSEC Technique: Traffic amplification

- A 50 byte query can lead to a 4037 byte response → typically factors of 50 are observed!
- Turn your spoofed 100Mbit/s DoS into a massive 5Gbit/s DoS!
  - Play with the big boys
  - Spread the load over multiple (hundreds!) of DNS servers
- Problem was well known, everybody talked about until it was no longer a problem (?!)

DNSSEC security?

- DNSSEC is of course about security
  - **Information** security
- Like OpenSSL
  - ...
- If we look at the past years of DNS security advisories, the majority has been about DNSSEC!
  - Helpful
- DNSSEC occupies hundreds of thousands of lines of new code

DNSSEC security?

- There have been three classes of DNSSEC security issues so far
  - Insufficient validation (everybody has suffered from this)
  - Denial of service due to algorithm misunderstandings (RIPE incident)
  - Classic buffer/stack overflows
- The middle one still haunts RIPE and means people are a bit scared about changing keys or algorithms
  - → you get hammered!

End-to-end DNSSEC

- Information security only works if the information is secured until it arrives in a ”safe place”
  - Oddly enough this safe place is called ”the browser” (oops)
- Current DNSSEC deployments are secure up to the ISPs resolver
  - From that point onwards, everything is insecured plaintext
  - Only an unsigned flag indicates an answer is 'secure'
- ”Last mile” is unsecured!

End-to-End DNSSEC

- Wow! So why are people pushing providers to ”do” DNSSEC?
  - No idea
- Solution right now is for everyone to run a validating resolver (would kill the internet)
- Better solutions mean that the ISP resolver ships all the signing proof to the stub resolver in the client PC (nice)
- Stub resolvers are limited though.. browsers themselves might do the validation though!