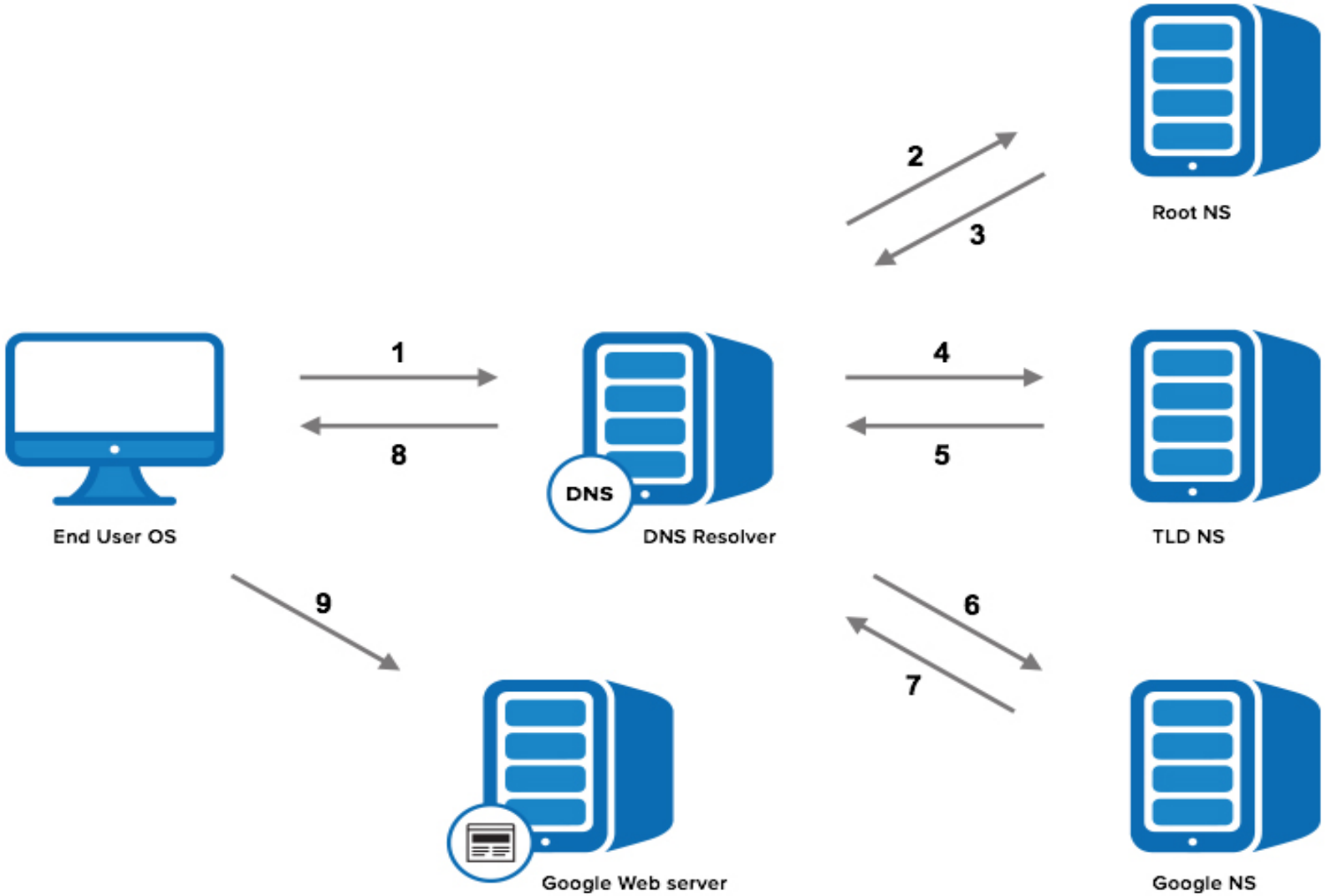
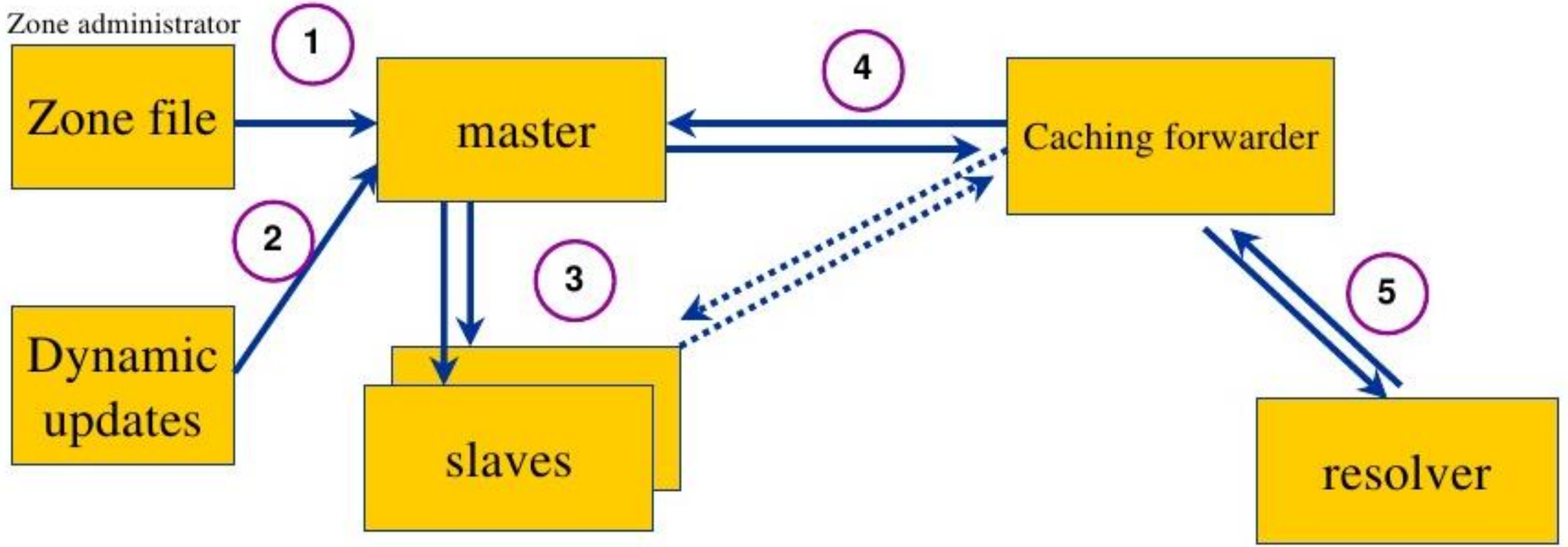
DNS Security: What, Why, How

Md. Mahedi Hasan
Technology Specialist (Innovation), BdREN, UGC
mahedi@bdren.net.bd

How DNS Works



DNS Data flow



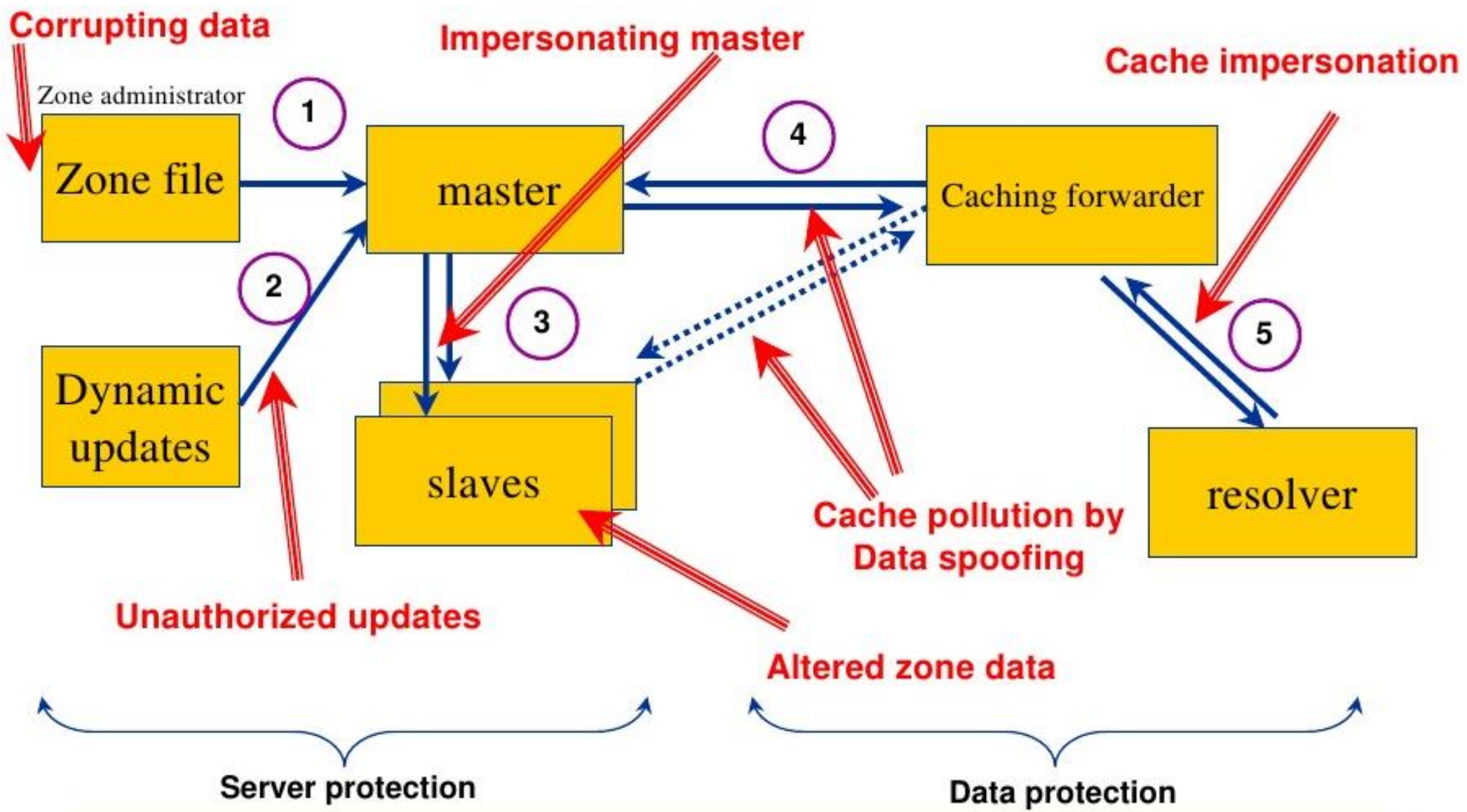
DNS Security ?

- The DNS is by its nature a public system and acts like a honeypot for the bad bees of the Internet world
- Unfortunately, the term DNSSEC has a bad reputation because of its perceived complexity and is frequently used to cover the whole topic of DNS security

DNS Security ?

- ***Administrative security:*** file permissions, server configuration, BIND configuration, and sandboxes (or chroot jails).
  - All the fancy cryptographic techniques in the world are useless if the base system is unstable or has world read-and-write privileges on all of the interesting files.
- ***Zone transfers:*** *Z*one transfers are essential to normal operation.
- ***Dynamic updates:*** Dynamic updates expose a master zone file to possible corruption, destruction, or poisoning.
- ***Zone integrity***

DNS Vulnerabilities



DNS Security Threats

Number	Area	Threat	Classification	Solutions
1	Zone files	File corruption (malicious or accidental)	Local	System administration
2	Dynamic updates	Unauthorized updates, IP address spoofing (impersonating update source)	Server-to-server	Network architecture, Transaction Signatures (TSIG), SIG(0), or disable
3	Zone transfers	IP address spoofing (impersonating update source)	Server-to-server	Network architecture, TSIG, or disable
4	Remote cache	Cache poisoning by IP spoofing, data interception, or a subverted master or slave	Server–client	DNSSEC
5	Resolver queries	Data interception, poisoned cache, subverted master or slave, local IP spoofing	Remote cache–client	DNSSEC

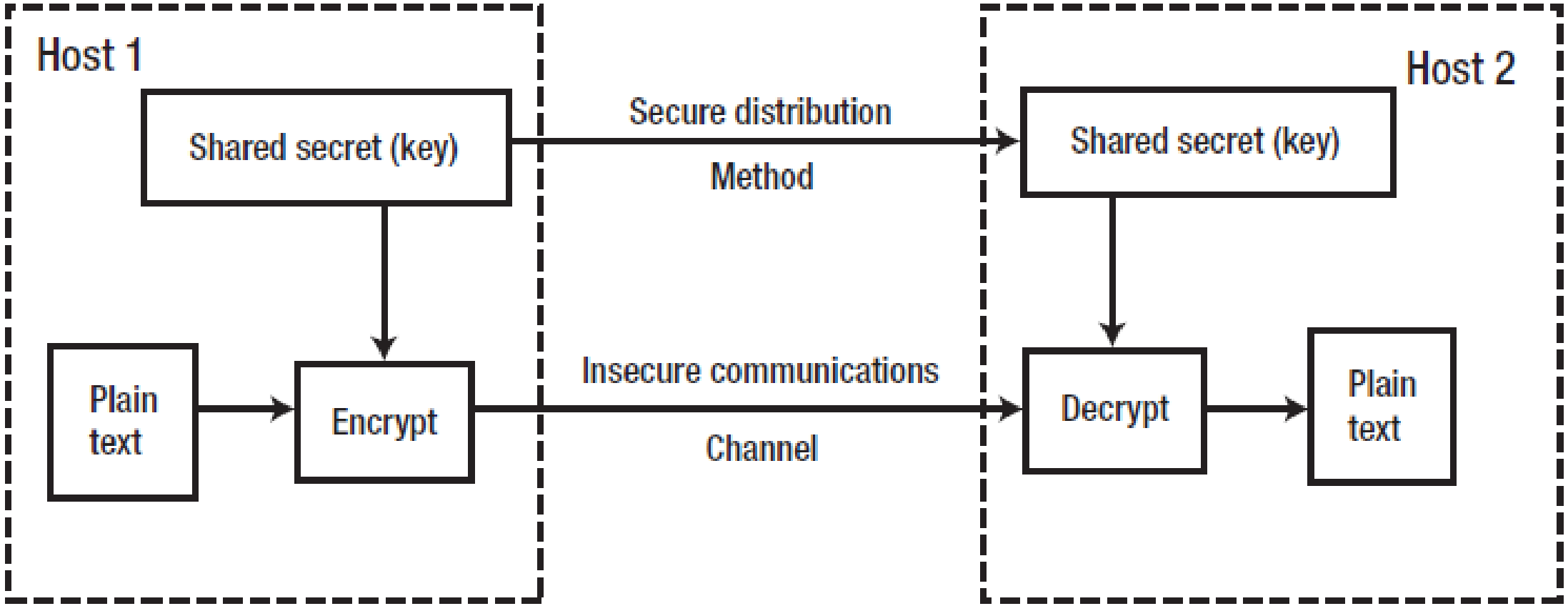
Stream the Log

```
// named.conf fragment
logging{
    channel normal_log {
        file "/var/log/named/normal.log" versions 3 size 2m;
        severity error;
        print-time yes;
        print-severity yes;
        print-category yes;
    };
    channel security_log { // streamed security log
        file "/var/log/named/security.log" versions 3 size 500k;
        severity info;
        print-time yes;
        print-severity yes;
        print-category yes;
    };
    category default{
        normal_log;
    };
    category security{
    security_log;
    };
};
....
```

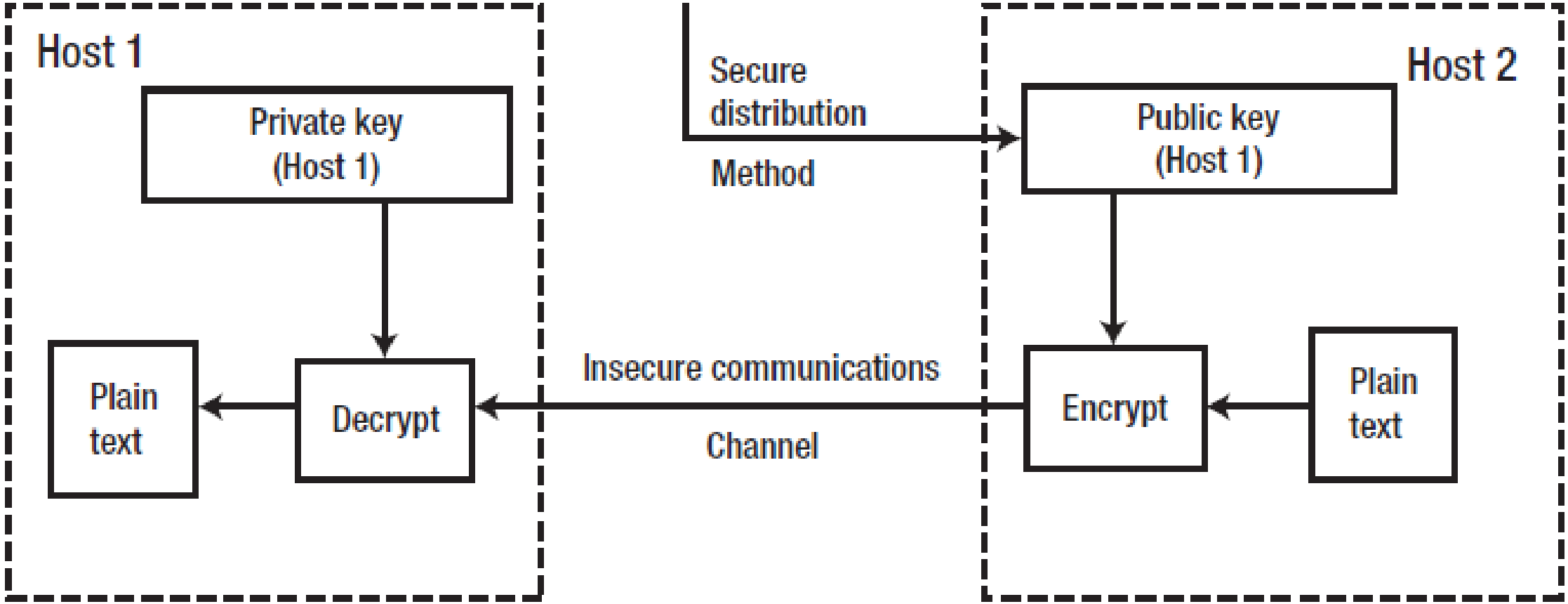
A Cryptographic Overview

- ***Confidentiality:*** Only the parties to the communication can understand the messages sent between the parties.
- ***Authentication:*** The data could only have come from a known source
- ***Data integrity:*** The data that is received by one party is the data that was sent by the other party

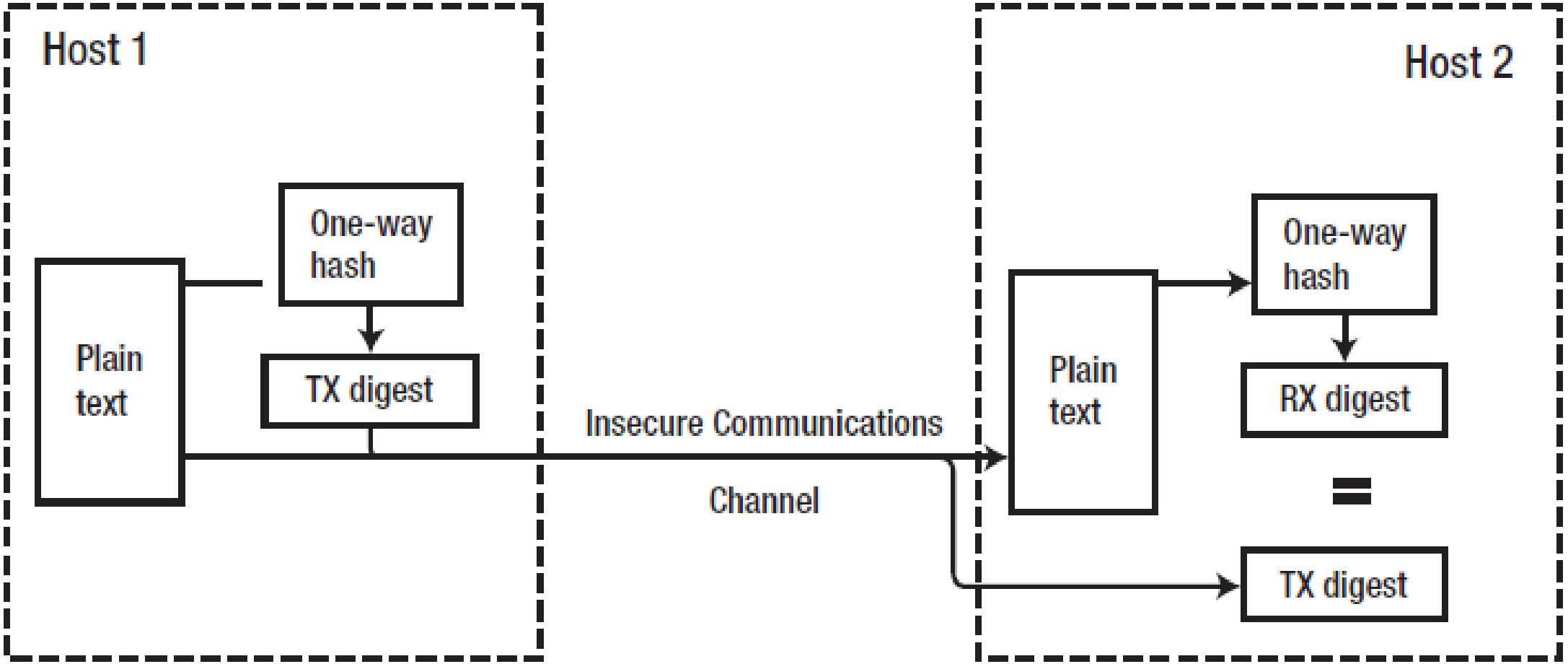
Symmetric Cryptography



Asymmetric Cryptography

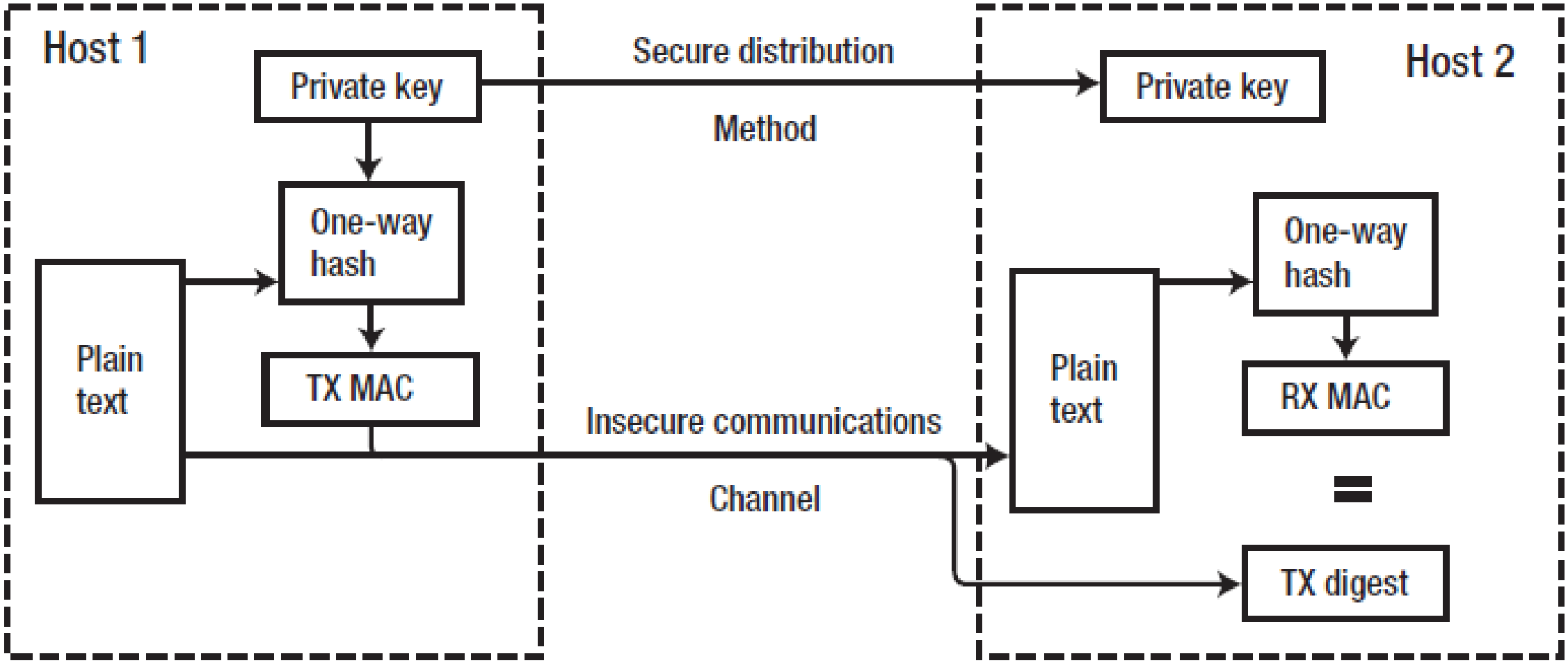


Message Digests



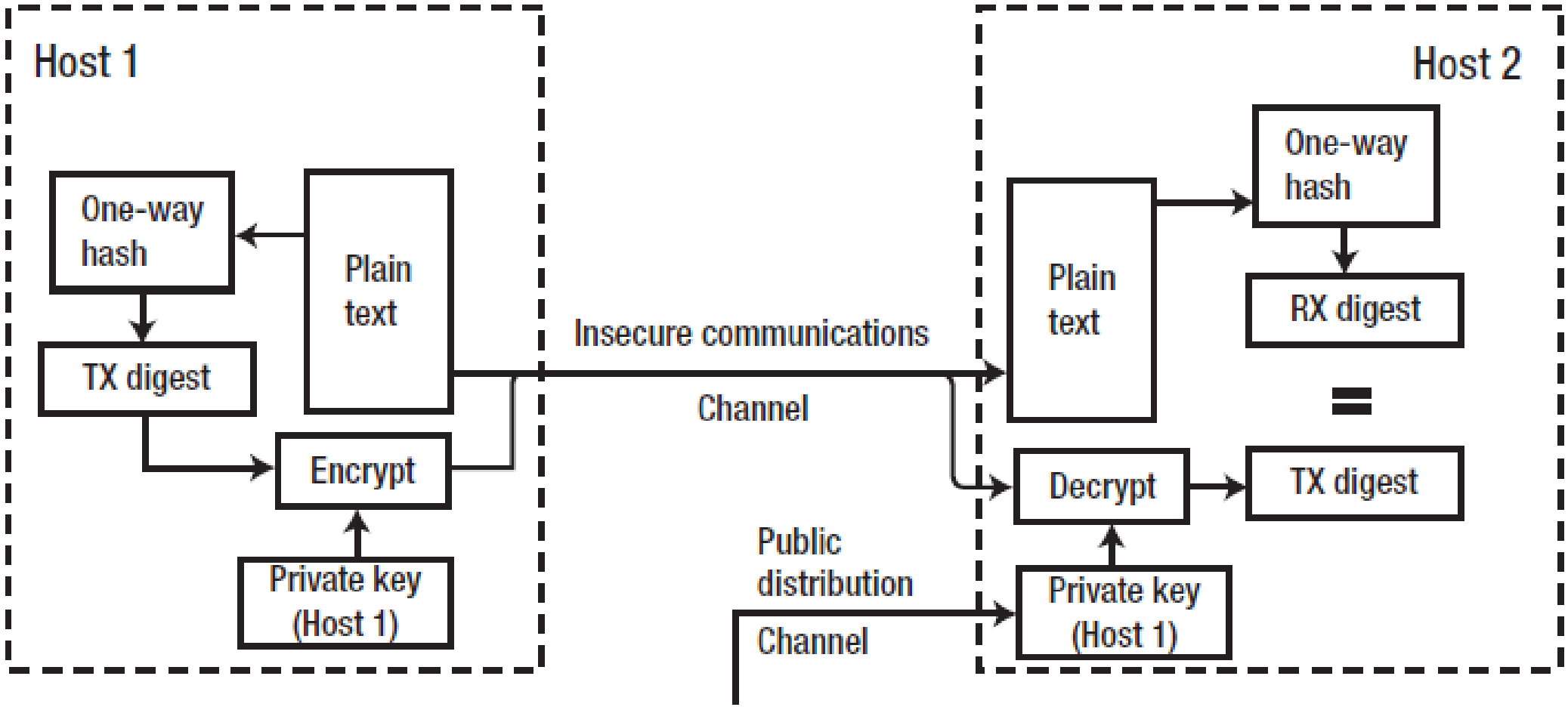
The most common forms of message digest are MD5, SHA-1, and increasingly SHA-2 (SHA-224, SHA-256, SHA-384, SHA- SHA-512)

Message Authentication Codes



The most common forms of MACs are HMAC-MD5, HMAC-SHA-1, and increasingly HMAC-SHA-256. MACs are used for TSIG secure operations in DNS.

Digital Signatures



The most common digital signature algorithms are RSA-MD5, RSA-SHA-1, RSA-SHA-256, and Digital Signature Architecture (DSA; a US government standard). Digital signatures are used in the DNS for SIG(0) secure transactions and for all DNSSEC transactions.

Securing Zone Transfers

- In most DNS configurations, zone transfers are essential.
- The default option in BIND is to allow zone transfers to any requesting host.
- If data should not be public, it should not be in the zone file on a public server.
- Simply securing zone transfers is not a solution to hiding data.
- The simplest way to secure zone transfers is through the use of IP addresses in BIND's named.conf file.

Deny All, Allow Selectively

```
options {
....
     allow-transfer {none;}; // no transfer by default
....
};
....
zone "example.com in{
....
     allow-transfer {10.0.1.2;}; // this host only
....
};
```

Authentication and Integrity of Zone Transfers

- **TSIG:**
- TSIG was defined in RFC 2845 and uses a single shared secret between the master and slave servers as part of a MAC.
- The key must be distributed to the slave locations by some secure process, such as fax, mail, courier, or secure e-mail, and it must be maintained securely at all the sites.
- If there is more than one slave server, either separate shared secrets may be used for each master-slave pair or a single shared secret may be used for all slaves.
- There is no change to the operational zone files when using the TSIG method; only the named.conf file is modified.

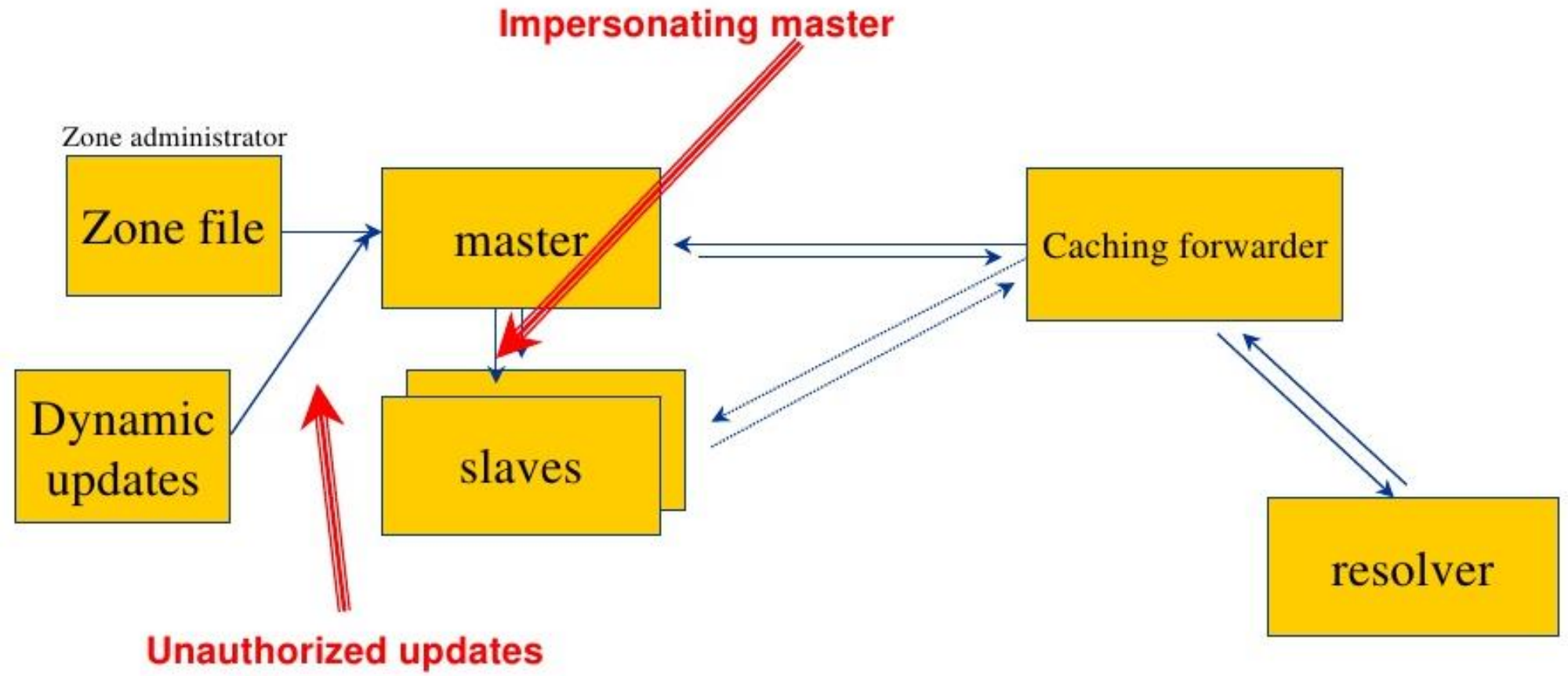
Authentication and Integrity of Zone Transfers

- **SIG(0):**
- SIG(0) was defined in RFC 2931 and uses a public-key system to generate a digital signature that both authenticates and ensures the integrity of the data involved in each transaction that includes zone transfers.
- However, there are no tools available with current BIND 9 releases to support SIG(0) for zone transfers.
- SIG(0) may be used with DDNS;

Authentication and Integrity of Zone Transfers

- **TKEY:**
- TKEY: The TKEY provides a method of securely exchanging shared-secret keys.
- The method is defined to support both the Diffie-Hellman algorithm and the Generic Security Services API (GSSAPI).
- However, the standard (RFC 2930) mandates that the exchange must be authenticated with either TSIG or SIG(0) methods.
- Consequently, it appears not to be widely implemented;

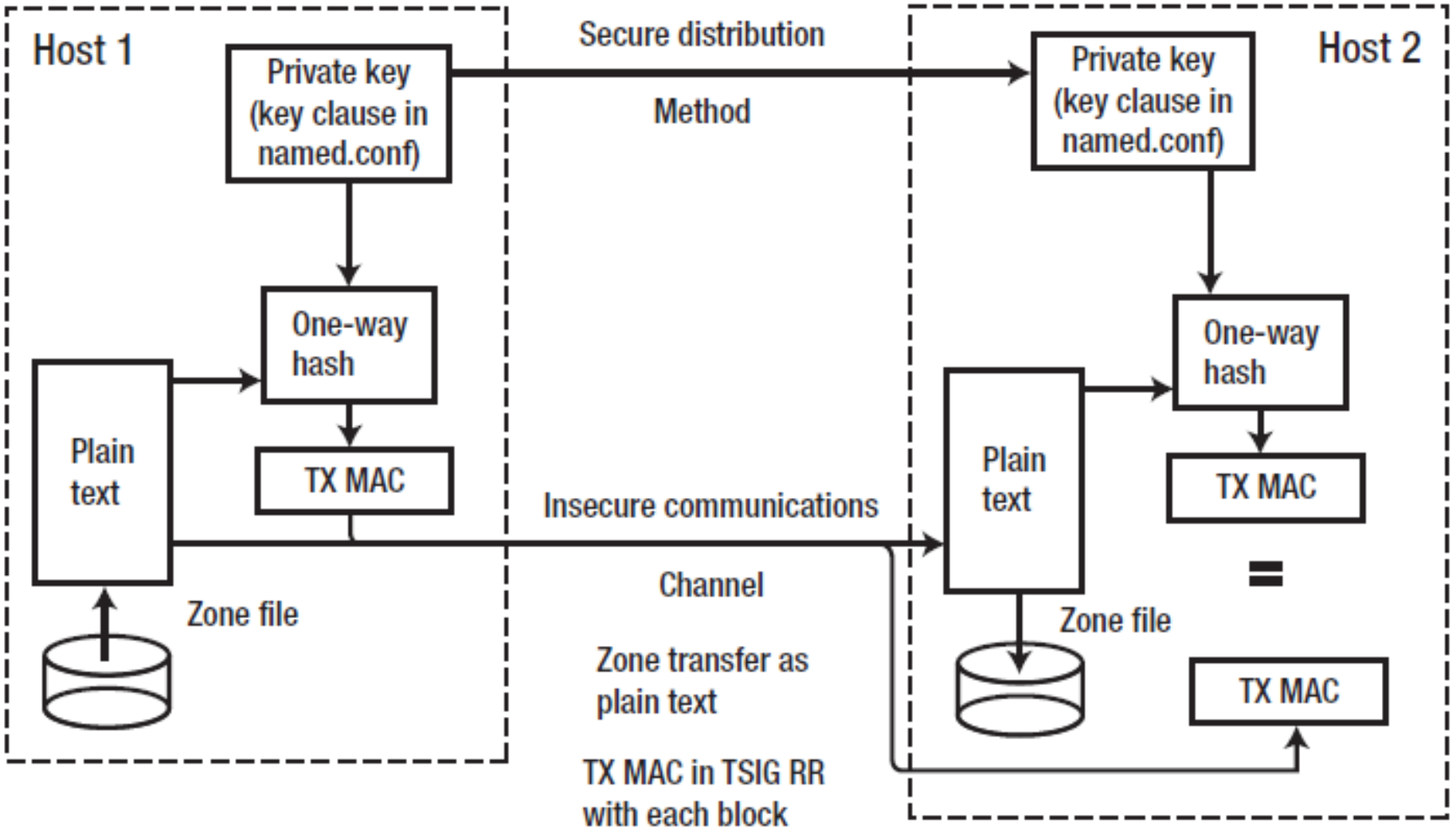
TSIG Protected Vulnerabilities



TSIG Configuration

- Transaction signatures (TSIG) use a MAC with a shared secret both to authenticate and ensure the data integrity of every transaction involved in zone transfers between the nominated slave and its master
- It is vital to keep in mind that shared-secret data is never placed in the DNS zone files.
- Instead, the shared secret is used by the two servers when exchanging data, such as a zone transfer

TSIG Configuration



DNSSEC

- When a name server receives the response to a query for, say, the A record of a web site, for instance, www.example.com, it can only hope that the data is correct.
- It has no way of proving that this is the case. In fact, it could have been duped or spoofed in a variety of ways, such as the query response may have been supplied from a poisoned zone file, or the query may have been intercepted and bad data substituted in the response.
- Another possibility is the query may have been redirected by a poisoned resolver cache to a bogus server for the domain in question, or the response could be perfectly valid, containing good data from the correct source.
- DNSSEC, originally defined in RFC 2535, was designed to eliminate the doubt involved in DNS query operations by providing verifiable certainty to suitably configured name servers.
- Significant efforts were expended over many years by multiple organizations, notably ISC (www.isc.org), Nlnetlabs (www.nlnetlabs.nl), some of the root-server operators (www.root-servers.org), and regional Internet registries (www.nro.net), to build and test secure
- DNSSEC (defined by RFCs 4033, 4034, and 4035 and augmented by 4470, 4509, 5011, and 5155) and constitutes a substantial enhancement to the original specifications.

DNSSEC History

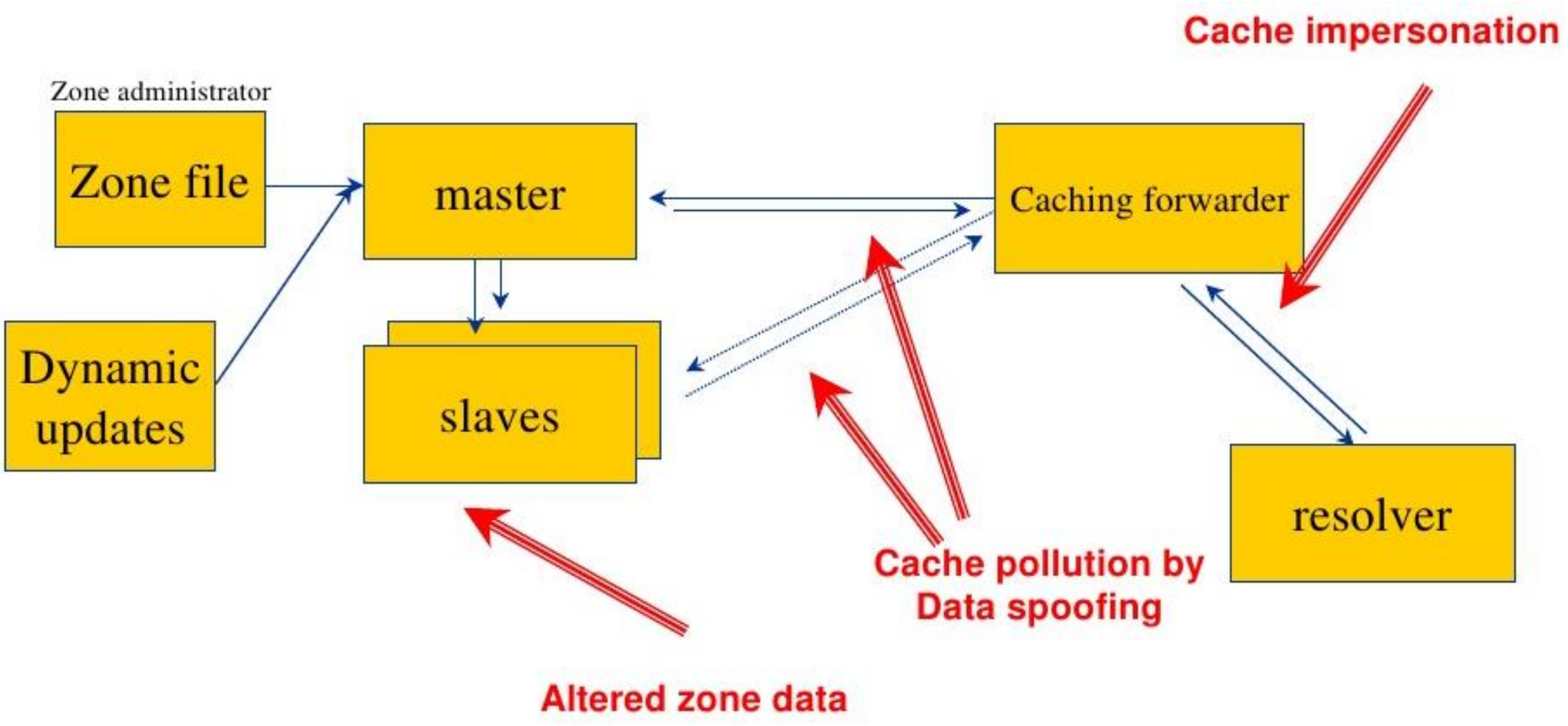
- 1990: Steven Bellovin discovers a major flaw in the DNS
- 1995: Bellovin publishes his research; DNSSEC (as it became later known) becomes a topic within IETF
- 1997: RFC 2065 (adding security extensions) was published
- 1998: Dan Kaminsky discovers some security flaw
- 1999: RFC 2535, the DNSSEC protocol, is published; BIND 9 developed to be DNSSEC-capable
- 2001: key handling in RFC2535 is causing operational problems

DNSSEC History (Cont.)

- 2005: Three new RFCs published to update RFC2535
  - RFC 4033 (DNS Security Introduction and Requirements)
  - RFC 4034 (Resource Records for DNS Security Extensions)
  - RFC 4035 (Protocol Modifications)
- 2005: In October, Sweden (.SE) becomes the first ccTLD to deploy DNSSEC
- 2008: new DNSSEC record created to address privacy concerns (RFC 5155)
- 2010 – In July 15, the root zone was signed – In July 29, .edu was signed – In December 9, .net was signed
- 2011: In March 31, .com was signed

https://wiki.tools.isoc.org/DNSSEC_History_Project

Vulnerabilities protected by DNSKEY / RRSIG / NSEC



Base DNSSEC Theory

- *Authenticate* that the data received could only have originated from the requested zone.
- *Verify the integrity of the data*: The data that was received at the querying resolver was the data that was sent from the queried named server. The data content is protected, not the communication channel.
- Verify that if a negative response (NXDOMAIN) was received to a host query, that the target record does not exist (called proof of nonexistence (PNE) and occasionally denial of existence).

DNSSEC hyper summary

- Data authenticity and integrity by signing the Resource Records Sets with private key
- Public DNSKEYs used to verify the RRSIGs
- Children sign their zones with their private key
  - Authenticity of that key established by signature/checksum by the parent (DS)
- Ideal case: one public DNSKEY distributed

DNSSEC Resource Records

- 3 Public key crypto related RRs
  - RRSIG = Signature over RRset made using private key
  - DNSKEY = Public key, needed for verifying a RRSIG
  - DS = Delegation Signer; 'Pointer' for building chains of authentication
- One RR for internal consistency
  - NSEC = Next Secure; indicates which name is the next one in the zone and which type codes are available for the current name
    - authenticated non-existence of data

DNSSEC Resource Records

- DNSKEY, RRSIG, and NSEC records provide mechanisms to establish authenticity and integrity of data
- DS record provides a mechanism to delegate trust to public keys of third parties

Thank You

?