SOFTWARE USER MANUAL (SUM): TRAINING, PROCEDURAL, AND DEVELOPMENT DOCUMENTATION

Step-by-Step DNS Security
Operator Guidance Document
(Version 0.5)
[*Using the BIND-9.3.0 (or later) distribution*]

31 August 2005

SUBMITTED BY

Sparta, Inc
7075 Samuel Morse Dr.
Columbia, MD 21046-3401

DNSSEC-Tools
Is your domain secure?

Contents

List of Figures

1 Introduction

DNS Security (DNSSEC) helps protect against DNS-spoofing attacks by providing origin authentication and integrity protection of DNS information. Proper maintenance of a DNSSEC-enhanced DNS zone is essential to protecting the domain's zone data.

This Step-by-Step DNS Security Operator Guidance document is intended for operations using the BIND-9.3.0 (or later) distribution. It will assist operators in gaining operational experience with DNSSEC. Some basic understanding of DNSSEC terms and concepts is required.

This document is meant to be a learning aid and is not intended to define policy in any form. Any implicit recommendations for key sizes, signature validity periods, and command line parameters are for illustration purposes ONLY and MUST NOT be used in production environments unless due-diligence has been taken to ensure that these values are acceptable within such environments. See [1] for suggestions on determining appropriate security characteristics.

This document was written as part of the DNSSEC-Tools project. The goal of this project is to create a set of documentation, tools, patches, applications, wrappers, extensions, and plug-ins that will help ease the deployment of DNSSEC-related technologies. For more information about this project and the tools that are being developed and provided, please see the DNSSEC-Tools project web page at:

http://www.dnssec-tools.org

1.1 Organization of this Document

The following operations are described in this document:

Section 2 Essential Preliminaries
This section contains essential instructions that must be followed before continuing with the rest of the document.

Section 3.1 Zone-Signing Key (ZSK) Generation
This section describes the procedure for creating new Zone Signing Keys; i.e., the keys used for signing zone data.

Section 3.2 Key-Signing Key (KSK) Generation
This section describes the procedure for creating new Key Signing Keys; i.e., keys that are used to sign the ZSKs in the apex keyset.

Section 3.3 Configuring and Serving a Signed Zone
This section describes the procedure for serving a signed zone file.

Section 3.4 Current ZSK Roll-Over

This section describes the procedure for rolling over an old ZSK. These steps should be used only if the older ZSK is known to have not been compromised.

Section 3.5 KSK Roll-Over

This section describes the procedure for rolling over an old KSK. These steps should be used only if the older KSK is known to have not been compromised.

Section 4.1 Signing a Zone with No Delegations

This section describes the procedure for signing a zone that has no delegations (no non-authoritative NS records) present in the zone file.

Section 4.2 Creating a Signed Delegation in a Child Zone

This section describes the activities that a child zone must perform in order to facilitate the creation of a signed delegation at the parent.

Section 5.1 Signing a Zone that Has Delegations

This section describes the procedure for signing a zone that has delegations (non-authoritative NS records) present in the zone file. The difference between this and section 3 lies in the additional communication involved between the parent and the child, as well as creation of the DS record in the parent zone, when delegations are present.

Section 5.2 Creating a Signed Delegation in a Parent Zone

This section describes the activities that a parent zone must perform in order to facilitate the creation of a signed delegation at the parent.

Section 6.1 KSK Roll-Over – KSK Compromise

This section describes the procedure for performing an emergency roll-over of the KSK when it is suspected or known to be compromised.

Section 6.2 ZSK Roll-Over – Current ZSK Compromise

This section describes the procedure for performing an emergency roll-over of the Current ZSK, when it is suspected or known to be compromised.

Section 6.3 ZSK Roll-Over – Published ZSK Compromise

This section describes the procedure for performing an emergency roll-over of the Published ZSK, when it is suspected or known to be compromised.

Section 6.4 ZSK Roll-Over – Published and Current ZSK Compromise

This section describes the procedure for performing an emergency roll-over of the Published and Current ZSKs, when both are suspected or known to be compromised.

Section 7.1 KSK Roll-Over – Parent Action During KSK Compromise

This section describes the actions that the parent zone must perform when it receives notification from the child about a KSK compromise, and before it publishes a DS value that points to the new KSK in the child.

These sections are followed by several appendices. These appendices contain useful information, such as checklists for the operations and pictorial illustrations for each of the operations described in this guide.

1.2 Identifying Relevant Steps

The following table summarizes the list of steps relevant to different kinds of zones. The columns marked with an 'X' for any row correspond to those operations with which the zone operator for that type of zone must be familiar.

Zone Profile	**Steps**													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
No signed delegations, parent is not signed	X	X	X		X			X	X	X	X	X	X	
No signed delegations, parent is signed	X	X	X		X	X		X	X	X	X	X	X	
Signed delegations, parent is not signed	X	X		X	X		X	X	X	X	X	X	X	X
Signed delegations, parent is signed	X	X		X	X	X	X	X	X	X	X	X	X	X

1.3 Key Concepts

ZSK - Zone-Signing Key
The key used to sign zone data.

KSK - Key-Signing Key
The key used to sign the ZSKs in the apex keyset.

Current ZSK
The Current ZSK is the key that is currently used to sign zone data.

Published ZSK
The Published ZSK is the key that is pre-published in the zone file as the future ZSK.

New ZSK
The New ZSK is the key that is scheduled to become the Published ZSK.

1.4 Conventions Used in this Document

One of the goals of this document is to self-contain DNS Security operations within sections and prevent constant cross-referencing between sections. Consequently, certain parts of the text are repeated throughout the document.

In particular, one might notice that zone SOA serial numbers may not change between sections. This should **not** be taken as an indication that the serial numbers do not need to change when the guide states that they should.

Text marked in bold represents text or commands entered by users within a given procedural step.

Underlined text, which can also be in bold, is a place-holder for actual run-time values. These values are either automatically generated or are values that are known to the user from some other step.

Additionally, the following typographical conventions are used in this document.

command	Command names
path	File and path names
URL	Web URLs
execution	Simple command executions

Longer sets of command sequences are given in this format:

```
$ cd /tmp [ENTER]
$ ls [ENTER]
$ rm -fr * [ENTER]
$
```

In most cases, output will not be displayed for given command sequences.

1.5 Acknowledgments

This document builds upon the procedures described in [1] for key roll-over techniques; the step-by-step instructions described in Sections 3.4 and 3.5 are meant to closely follow the recommendations given by that document. Early versions of this guide were reviewed and critiqued by SAIC, Inc./Quotient, Inc., including Rip Loomis and Rob Payne. Their contributions are much appreciated.

1.6 Comments

Please send any comments and corrections to developers@dnssec-tools.org.

2 Essential Preliminaries

The following sections must be read before proceeding with the rest of this guide.

The steps in Sections 2.1 and 2.2 **MUST** be taken prior to following any other steps. Section 2.3's discussion of this guides use of zone file serial numbers **MUST** be understood prior to following any other steps. Failure to do any of these three could affect the security of your zone.

Section 2.4 describes *key-tags* tables, which are used to manage DNSSEC information about encryption keys and signed zones. This section should be understood prior to reading the remainder of the document.

2.1 Check for Randomness

Key generation and zone signing require random data to create strong cryptographic material. The *dnssec-keygen* and *dnssec-signzone* commands default to using random data from **/dev/random**. Use this test to verify that **/dev/random** will provide data when requested:

```
$ dd if=/dev/random bs=2 count=10 | od -x [ENTER]
...
$
```

The above command checks if **/dev/random** is able to provide random data when queried; it does not check to see that the data provided is truly random.

If this command provides data immediately, then **/dev/random** will provide the data you need. If it hangs, then *dnssec-keygen* and *dnssec-signzone* won't be able to retrieve random data from **/dev/random**.

If this check for randomness fails, pseudorandom numbers can be used instead. However, using pseudorandom numbers significantly affects the quality of the crypto material. A more appropriate measure would be to run the key-generation and zone-signing operations on a different system that has **/dev/random** and the ability to generate good random data.

2.2 Check for Correct Version of BIND

BIND version 9.3.0 (or later) is **required** for use with this document. Before following any of these instructions, you **must** ensure that you have the correct version of BIND.

The BIND version may be verified by checking the version of *dnssec-keygen*:

```
$ dnssec-keygen -h [ENTER]
Usage:
dnssec-keygen -a alg -b bits -n type [options] name
Version: 9.3.0
......
$
```

If the version is incorrect, you must install the correct version of BIND before proceeding.

2.3 Zone File Serial Numbers

This guide contains a number of instructions to update a zone file. In each, it is indicated that the zone file's serial number must be updated. The actual serial numbers used will depend on each installation's preferred method of using serial numbers. The method used in this document is YYYYMMDDNN, where:

- **YYYY** - year
- **MM** - month
- **DD** - day
- **NN** - incrementing number

For example, the third zone file change on January 9, 2005, would be given as 2005090103.

The serial numbers used in this guide are *generally* increasing throughout the document. More importantly, however, the numbers used within each section are *always* increasing.

Some sections of this guide direct you to perform steps in other sections. Regardless of the serial numbers given in these steps, your zone file's serial numbers **must** increase with each modification or your changes will not be seen.

2.4 Key-Tags Tables

Data about results of the different steps described in this document must be retained for use in other steps. These data include such things as the names of generated keys, key status, and the date a zone was signed. While these data may be kept in whatever form an administrator finds most convenient, this guide stores them in a *key-tags table.*

A key-tags table stores a zone name, the zone's ZSK keys, the zone's KSK keys, and the zone's expiration date. The following is a template of how the key-tags table is used in this document. Data about the zone's KSK, Current ZSK, Published ZSK, and New ZSK are stored here:

Zone	**ZSK**				**KSK**				**Exp**
	Tag	Size	Creat	S	Tag	Size	Creat	S	
zone.name	zsktag-cur	512	date	C	ksktag	1024	date	C	
	zsktag-pub	512	date	P					
	zsktag-new	512	date	N					

The *-cur*, *-pub*, and *-new* suffixes used in this key-tags table and in the rest of this guide are used for illustration purposes only. They are not intended to be used in the key-tags table for an operational zone. Similarly, for convenience and identification purposes the tags are retained with particular keys even when the status changes.

The following is an example key-tags table containing non-template data. There are entries for two zones: *example.com* and *example.net.* There are entries for a example.com's Current KSK, New KSK, Current ZSK, Published ZSK, and New ZSK. example.net's entries are for its Current KSK, Current ZSK, and Published ZSK.

Zone	**ZSK**				**KSK**				**Exp**
	Tag	Size	Creat	S	Tag	Size	Creat	S	
example.com	32267	512	8/15/05	C	65037	1024	9/21/04	C	9/21/05
	40379	512	8/20/05	P	14895	1024	9/21/05	N	
	58406	512	8/25/05	N					
example.net	30449	512	7/24/05	C	25384	1024	1/09/05	C	9/27/05
	38715	512	8/27/05	P					

As stated above, this information may be kept in whatever form each administrator finds convenient. Some administrators may want to have separate key-tags tables for each zone for which they are responsible; others may wish to maintain all their zones in a single table.

A blank template key-tags table may be found in Appendix B. Additional versions (PostScript, Excel spreadsheet) may be found on **http://www.dnssec-tools.org**.

3 Normal Operations for All Zones

This section describes those normal DNSSEC operations which are relevant for all zones. These operations are:

- Zone-Signing Key (ZSK) Generation
- Key-Signing Key (KSK) Generation
- Configuring and Serving a Signed Zone
- Current ZSK Roll-Over
- KSK Roll-Over

3.1 Zone-Signing Key (ZSK) Generation

This section provides the steps required to generate a new Zey-Signing Key (ZSK). See Figure 1.

3.1.1 Generate the Key

Use the *dnssec-keygen* command to generate a key 512 bits long.

```
$ dnssec-keygen -a RSASHA1 -b 512 -n ZONE zone.name
 [ENTER]
 Kzone.name.+005+zsktag
$
```

The process may take a few minutes to return its results. If the process appears to have stalled, run the command using a pseudo-random number generator as follows:

```
$ dnssec-keygen -r /dev/urandom -a RSASHA1 -b 512
 -n ZONE zone.name [ENTER]
 Kzone.name.+005+zsktag
$
```

Two files are output by *dnssec-keygen*:

- Private key contained in **Kzone.name.+005+zsktag.private**
- Public key contained in **Kzone.name.+005+zsktag.key**

zone.name - the name of the zone (e.g., example.com)
zsktag - the key identifier (e.g., 57011)

You must note this number in the key-tag table as you walk through this document. This number is automatically generated and should not be changed. The key id will be the only field in the filename that changes as you rotate keys, so it must be tracked.

This document uses RSASHA1 as the cryptographic algorithm, which is represented by the "005" in the key name's algorithm field.

3.1.2 Update the Key-Tags Table

Keep a record of the key-tags that currently refer to ZSKs.

Zone	**ZSK**				**KSK**				**Exp**
	Tag	Size	Creat	S	Tag	Size	Creat	S	
zone.name	zsktag	512	date						

Leave the status field (S) empty for now.

3.1.3 Store Separately the Private Keys for ZSKs and KSKs

Storing the private keys for ZSKs and KSKs off-line is considered a good security practice. ZSK and KSK separation lessens the operational burden during zone-key compromise, but only if the KSK is still considered safe for use. In cases where a signed zone is also updated via dynamic updates, the ZSK will need to be on-line and available to the name server process. To avoid a common vulnerability point (such as compromise of the system on which these keys reside), store the KSKs in a place that is considered safer.

3.2 Key-Signing Key (KSK) Generation

This section provides the steps required to generate a new Key-Signing Key (KSK). See Figure 1.

3.2.1 Generate the Key

Use the *dnssec-keygen* command to generate a key 1024 bits long. Key size is a rough measure of strength; KSKs are generally made stronger than ZSKs.

```
$ dnssec-keygen -a RSASHA1 -b 1024 -n ZONE -f KSK zone.name
[ENTER]
Kzone.name.+005+ksktag
$
```

The process may take a few minutes to return its results. If the process appears to have stalled, run the command using a pseudo-random number generator as follows:

```
$ dnssec-keygen -r /dev/urandom -a RSASHA1 -b 1024 -n ZONE -f KSK
zone.name [ENTER]
Kzone.name.+005+ksktag
$
```

Two files are output by *dnssec-keygen*:
- Private key contained in **Kzone.name.+005+ksktag.private**
- Public key contained in **Kzone.name.+005+ksktag.key**

zone.name - the name of the zone (e.g., example.com)
ksktag - the key identifier (e.g., 24818)

You must note this number in the key-tag table as you walk through this document. This number is automatically generated and should not be changed. The key id will be the only field in the filename that changes as you rotate keys, so it must be tracked.

This document uses RSASHA1 as the cryptographic algorithm, which is represented by the "005" in the key name's algorithm field.

3.2.2 Update the Key-Tags Table

Keep a record of the key-tags that currently refer to KSKs.

Zone	**ZSK**				**KSK**				**Exp**
	Tag	Size	Creat	S	Tag	Size	Creat	S	
zone.name					ksktag	1024	date		

Leave the status field (S) empty for now.

3.2.3 Store Separately the Private Keys for ZSKs and KSKs

Storing the private keys for ZSKs and KSKs off-line is considered a good security practice. ZSK and KSK separation lessens the operational burden during zone-key compromise, but only if the KSK is still considered safe for use. In cases where a signed zone is also updated via dynamic updates, the ZSK will need to be on-line and available to the name server process. To avoid a common vulnerability point (such as compromise of the system on which these keys reside), store the KSKs in a place that is considered safer.

3.3 Configuring and Serving a Signed Zone

Several configuration files must be modified in order to serve a signed zone. Follow the steps below to configure your name server and have it start serving your signed zone.

named.conf is the name of the configuration file used in these examples. The configuration file may vary according to the needs of the administrator.

3.3.1 Add the Signed Zone to the Name Server Configuration File

The name of the signed zone file must be added to the name server's configuration file. For the zone whose name is zone.name, do the following:

```
$ vi named.conf [ENTER]
 ...
 zone "zone.name." {
              type master;
              file "zonefile.signed";
 };
 ...
$
```

3.3.2 Enable DNSSEC

Add the *dnssec-enable yes;* option to the **named.conf** file.

```
$ vi named.conf [ENTER]
 ...
 options {
              ...
              dnssec-enable yes;
 };
 ...
$
```

3.3.3 Check the Name Server Configuration File for Errors

You must ensure that the configuration file modifications were performed correctly. The *named-checkconf* command will perform this verification. No output indicates that all is well with the zone.

```
$ named-checkconf named.conf [ENTER]
$
```

3.3.4 Reload the Zone

The *rndc* command will reload the name server configuration files and the zone contents. The name server process is assumed to be already running.

```
$ rndc reload zone.name [ENTER]
$
```

3.3.5 Check that the Zone Loaded Properly

Confirm that the SOA serial number of the zone corresponds to the most recent value.

```
$ dig @server-IP-address SOA zone.name [ENTER]
 ; <<>> DiG 9.3.0 <<>> ...
 ...
 ;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0
 ...
 ;;ANSWER SECTION
 zone.name          3600      IN        SOA      servername contact (
                                         2005092101 ; This should be the most
                                                    ; recent value.
                                                    ; This value will most
                                                    ; likely be different in
                                                    ; your zone file.
 ...
            )
...
$
```

3.4 Current ZSK Roll-Over

This section gives the steps necessary for the pre-publish scheme for ZSK roll-over. The alternative, the double-signature method, is used for rolling over KSKs. Double signatures for records signed by the ZSK can increase the size of the zone many times. The pre-publish scheme, although requiring more steps for the roll-over, does not suffer from this problem. The size argument does not apply during KSK roll-over since the DNSKEY RRset is the only record doubly signed by the KSK.

See Figure 2.

3.4.1 Ensure that Sufficient Time has Elapsed Since the Last Roll-Over

The time between roll-overs has to be at least twice the maximum zone TTL period. This is the largest TTL in the entire zone file multiplied by two.

3.4.2 Sign Zone with the KSK and Published ZSK

Follow steps 4.1.3 – 4.1.7 if the zone does no delegation. Follow steps 5.1.3 – 5.1.8 if the zone does delegation. The ZSK used in the signing process in Section 4.1.5 or 5.1.5 must be the key that is marked as the Published key (P) in the key-tag table. The KSK used as input to *dnssec-signzone* does not change, so the keyset does not change and does not have to be re-sent to the parent.

Record the signature expiry date in the key-tag table.

Zone	**ZSK**				**KSK**				**Exp**
	Tag	Size	Creat	S	Tag	Size	Creat	S	
zone.name	zsktag-cur	512	date	C	ksktag	1024	date	C	date
	zsktag-pub	512	date	P					

3.4.3 Reload the Zone

The *rndc* command will reload the name server configuration files and the zone contents. The name server process is assumed to be already running.

```
$ rndc reload zone.name [ENTER]
$
```

3.4.4 Wait for Old Zone Data to Expire from Caches

Wait at least twice the maximum zone TTL period for the old zone data to expire from name server caches. This is the largest TTL in the entire zone file multiplied by two. This will also allow the new data to propagate.

3.4.5 Generate a New ZSK

Generate a new ZSK, as described in section 3.1. Update the key-tag table with the new ZSK, and set its status to New (N).

Zone	**ZSK**				**KSK**				**Exp**
	Tag	Size	Creat	S	Tag	Size	Creat	S	
zone.name	zsktag-cur	512	date	C	ksktag	1024	date	C	date
	zsktag-pub	512	date	P					
	zsktag-new	**512**	**date**	**N**					

3.4.6 Modify the Zone File

The zone file must be modified to account for the key changes. The Current ZSK must be deleted and the New ZSK must be added. Also, the SOA serial number must be changed so that the zone file's new contents will be recognized.

```
$ vi zonefile [ENTER]
zone.name          IN        SOA       servername contact (
                                            2005092102 ; Increase current value by 1.
                                                       ; This value may be different
                                                       ; in your zone file.
                    ...
 )
...
;; ksk
$INCLUDE "/path/to/Kzone.name.+005+ksktag.key"
~~;; cur zsk~~
~~$INCLUDE "/path/to/Kzone.name.+005+zsktag-cur.key"~~
;; pub zsk
$INCLUDE "/path/to/Kzone.name.+005+zsktag-pub.key"
;; new zsk
$INCLUDE "/path/to/Kzone.name.+005+zsktag-new.key"
...
```

```
$
```

3.4.7 Update the Key-Tags Table

Update the key-tags table to reflect the changed key status. Delete the old Current ZSK. Change the status of the Published ZSK to Current. Change the status of the New ZSK to Published.

Zone	**ZSK**				**KSK**				**Exp**
	Tag	Size	Creat	S	Tag	Size	Creat	S	
zone.name	~~**zsktag-cur**~~	~~**512**~~	~~**date**~~	~~**C**~~	ksktag	1024	date	C	date
	zsktag-pub	512	date	~~**P**~~ **C**					
	zsktag-new	512	date	~~**N**~~ **P**					

3.4.8 Sign the Zone with the KSK and Current ZSK

Follow the steps 4.1.3 – 4.1.7 if the zone does no delegation. Follow the steps 5.1.3 – 5.1.8 if the zone does delegation. The ZSK used in the signing process in Section 4.1.5 or 5.1.5 must be the key that is marked as the Current key (C) in the key-tag table (this was the older Published key.) The KSK used as input to *dnssec-signzone* does not change, so the keyset does not change and does not have to be re-sent to the parent.

Record the signature expiry date in the key-tag table.

Zone	**ZSK**				**KSK**				**Exp**
	Tag	Size	Creat	S	Tag	Size	Creat	S	
zone.name	zsktag-pub	512	date	C	ksktag	1024	date	C	date
	zsktag-new	512	date	P					

3.4.9 Reload the Zone

The *rndc* command will reload the name server configuration files and the zone contents. The name server process is assumed to be already running.

```
$ rndc reload zone.name [ENTER]
$
```

3.4.10 Dispose of the Old Zone Key

Delete the old ZSK's *.private* and *.key* files.

3.5 KSK Roll-Over

This section gives the steps necessary for the double-signature scheme for KSK roll-over. The alternative, the pre-publish method, is used for rolling over ZSKs. Double signatures for records signed by the ZSK can increase the size of the zone many times. The pre-publish scheme, although requiring more steps for the roll-over, does not suffer from this problem. The size argument does not apply during KSK roll-over since the DNSKEY RRset is the only record doubly signed by the KSK.

See Figure 3.

3.5.1 Ensure that Sufficient Time has Elapsed Since the Last Roll-Over

The time between roll-overs has to be at least twice the maximum zone TTL period. This is the largest TTL in the entire zone file multiplied by two.

3.5.2 Generate a New KSK

Generate a new KSK, as described in section 3.2.

Zone	**ZSK**				**KSK**				**Exp**
	Tag	Size	Creat	S	Tag	Size	Creat	S	
zone.name	zsktag-cur	512	date	C	ksktag	1024	date	C	date
	zsktag-pub	512	date	P	**ksktag**	**1024**	**date**	**P**	

3.5.3 Modify the Zone File

The zone file must be modified to account for the new KSK. The SOA serial number also must be changed so that the zone file's new contents will be recognized.

```
$ vi zonefile [ENTER]
zone.name IN SOA servername contact (
                                        2005092103 ; Increase current value by 1.
                                                   ; This value may be different
                                                   ; in your zone file.
             ...
 )
...
;; cur ksk
```

```
$INCLUDE “/path/to/Kzone.name.+005+ksktag.key”
;; new ksk
$INCLUDE “/path/to/Kzone.name.+005+ksktag-pub.key”
;; cur zsk
$INCLUDE “/path/to/Kzone.name.+005+zsktag-cur.key”
;; pub zsk
$INCLUDE “/path/to/Kzone.name.+005+zsktag-pub.key”
...
$
```

3.5.4 Re-Sign the Zone DNSKEY RRset with the Current and Published KSK

ZSKs sign the zone data, whereas KSKs sign the RRset for all DNSKEYs recognized by the zone. There is no direct way to create the signed DNSKEY RRset for the zone; it is only formed as a by-product of the *dnssec-signzone* operation.

Follow steps 4.1.3 – 4.1.7 if the zone does no delegation. Follow steps 5.1.3 – 5.1.8 if the zone does delegation. The ZSK used in the signing process in Section 4.1.5 or 5.1.5 must be the key that is marked as the Current key (C) in the key-tag table. Both Current KSK and the Published KSK must be simultaneously included in the *dnssec-signzone* operation (by using two *-k* options).

Record the signature expiry date in the key-tag table.

Zone	**ZSK**				**KSK**				**Exp**
	Tag	Size	Creat	S	Tag	Size	Creat	S	
zone.name	zsktag-cur	512	date	C	ksktag-cur	1024	date	C	**date**
	zsktag-pub	512	date	P	ksktag-pub	1024	date	P	

Although the keyset has changed, it **must not** be sent to the parent yet.

3.5.5 Reload the Zone

The *rndc* command will reload the name server configuration files and the zone contents. The name server process is assumed to be already running.

```
$ rndc reload zone.name [ENTER]
$
```

3.5.6 Wait for Old DNSKEY RRset to Expire from Caches

Wait at least twice the maximum zone TTL period for the old DNSKEY RRset to expire from name server caches. This is the largest TTL in the entire zone file multiplied by two. This will also allow the new data to propagate.

3.5.7 Modify the Zone File

The zone file must be modified to account for the key changes. The Current ZSK must be deleted and the New ZSK must be added. Also, the SOA serial number must be changed so that the zone file's new contents will be recognized.

```
$ vi zonefile [ENTER]
zone.name IN SOA servername contact (
                                        2005092104 ; Increase current value by 1.
                                                   ; This value may be different
                                                   ; in your zone file.
          ...
 )
...
~~;; cur ksk~~
~~$INCLUDE "/path/to/Kzone.name.+005+ksktag-cur.key"~~
;; new ksk
$INCLUDE "/path/to/Kzone.name.+005+ksktag-pub.key"
;; pub zsk
$INCLUDE "/path/to/Kzone.name.+005+zsktag-pub.key"
;; new zsk
$INCLUDE "/path/to/Kzone.name.+005+zsktag-new.key"
...
$
```

3.5.8 Re-Sign the Zone DNSKEY RRset with the Current and Published KSK

ZSKs sign the zone data, whereas KSKs sign the RRset for all DNSKEYs recognized by the zone. There is no direct way to create the signed DNSKEY RRset for the zone; it is only formed as a by-product of the *dnssec-signzone* operation.

Follow steps 4.1.3 – 4.1.7 if the zone does no delegation. Follow steps 5.1.3 – 5.1.8 if the zone does delegation. The ZSK used in the signing process in Section 4.1.5 or 5.1.5 must be the key that is marked as the Current key (C) in the key-tag table. Both Current KSK and the Published KSK must be simultaneously included in the *dnssec-signzone* operation (by using two *-k* options).

3.5.9 Update the Key-Tags Table with the Latest KSK

Delete the Current KSK and change the status of the new KSK from Published (P) to Current (C). Record the signature expiry date in the key-tag table.

Zone	**ZSK**				**KSK**				**Exp**
	Tag	Size	Creat	S	Tag	Size	Creat	S	
zone.name	zsktag-cur	512	date	C	~~**ksktag-cur**~~	~~**1024**~~	~~**date**~~	~~**C**~~	**date**
	zsktag-pub	512	date	P	ksktag-pub	1024	date	~~C~~ **P**	

3.5.10 Perform Steps in Section 4.2 if this Zone is a Secure Delegation from Another Zone

The keyset generated in 3.5.8 contains only the new KSK. This keyset must be sent to the parent in order to complete the secure delegation.

3.5.11 Reload the Zone

The *rndc* command will reload the name server configuration files and the zone contents. The name server process is assumed to be already running.

```
$ rndc reload zone.name [ENTER]
$
```

3.5.12 Dispose of the Old KSK

Delete the old KSK's *.private* and *.key* files.

4 Normal Operations for Child Zones

This section describes those normal DNSSEC operations which are relevant for child zones. These operations are:

- Signing a Zone that Has No Delegations
- Creating a Signed Delegation in a Child Zone

4.1 Signing a Zone with No Delegations

A zone needs to be re-signed when any change is made to it. Steps 4.1.1 and 4.1.2 must be followed if the zone has not been previously signed. Steps 4.1.3 - 4.1.7 must be followed when re-signing a zone file that has no delegations.

4.1.1 Generate Two ZSKs and One KSK

Follow the steps in Section 3.1 for generating the ZSKs and steps in Section 3.2 for generating the KSK.

Designate one of the two ZSKs as the Current (C) zone signing key and use it to sign the zone data; designate the other as the Published (P) key, which is for future use following a ZSK roll-over. Set the status of each of these keys in the column marked S.

There is only one KSK and this is used to sign the zone apex keyset; mark its status as Current (C). The Published ZSK should be kept more safely[1] than the Current ZSK. The idea is that the Published ZSK can be easily rolled in even if the Current ZSK is compromised (the Current ZSK may have to be kept on-line in some circumstances).

If the KSK has been stored in a more secure location (off-line, more highly protected directory, etc.) then it might be a good idea to store the Published ZSK in the same secure location.

Zone	**ZSK**				**KSK**				**Exp**
	Tag	Size	Creat	S	Tag	Size	Creat	S	
zone.name	zsktag-cur	512	date	C	ksktag	1024	date	C	
	zsktag-pub	512	date	P					

4.1.2 Modify the Zone File

The zone file must be modified to account for the new keys. Add lines to include the KSK and the ZSKs in the zone file. Also, the SOA serial number must be changed so that the zone file's new contents will be recognized.

```
$ vi zonefile [ENTER]
zone.name IN SOA servername contact (
```

[1] It would be a good idea for an operator to apply increased protection mechanisms (physical, file permissions and ownership, network, etc.) to the Published ZSK than are used for the Current ZSK.

```
                                        2005092105 ; Increase current value by 1.
                                                   ; This value may be different
                                                   ; in your zone file.
            ...
    )
    ...
    ;; ksk
    $INCLUDE “/path/to/Kzone.name.+005+ksktag.key”
    ;; cur zsk
    $INCLUDE “/path/to/Kzone.name.+005+zsktag-cur.key”
    ;; pub zsk
    $INCLUDE “/path/to/Kzone.name.+005+zsktag-pub.key”
    ...
    $
```

4.1.3 Check the Unsigned Zone File for Errors

Ensure that the unsigned zone file was modified correctly.

```
$ named-checkzone zone.name zonefile [ENTER]
 zone zone.name/IN: loaded serial SerialNumber
 OK
$
```

4.1.4 Check Permissions and Ownership on ZSK and KSK Files

The key files must be readable by the *dnssec-signzone* tool and the name server's user id.

4.1.5 Sign the Zone

Use the *dnssec-signzone* command to sign the zone file.

```
$ dnssec-signzone -k /path/to/Kzone.name.+005+ksktag.key
 -o zone.name -e +2592000 zonefile
 /path/to/Kzone.name.+005+zsktag-cur.key [ENTER]
 zonefile.signed
```

```
$
```

Signature generation may take a few minutes to complete, depending on the size of the zone file. If the above operation appears to be unresponsive for an unreasonable length of time, use pseudorandom numbers (using the *-p* option) instead.

```
$ dnssec-signzone -k /path/to/Kzone.name.+005+ksktag.key
 -o zone.name -p -e +2592000 zonefile
 /path/to/Kzone.name.+005+zsktag-cur.key [ENTER]
 zonefile.signed
$
```

Three files are created by *dnssec-signzone*:

- Signed zone file in zonefile.signed. The *.signed* suffix is appended by default.
- Keyset file in keyset-zone.name. May have to be sent to the parent zone; see Section 4.2.
- DS-set file in dsset-zone.name. Used to verify that the correct DS record was generated at the parent; see Section 4.2.

This *dnssec-signzone* command generates signatures for the records that are valid for 30 days (2,592,000 seconds) from the current time. This is offset by -1 hour to account for clock skew between the name server and DNSSEC validators.

4.1.6 Check the Signed Zone File for Errors

Ensure that the signed zone file was modified correctly.

```
$ named-checkzone zone.name zonefile [ENTER]
 zone zone.name/IN: loaded serial SerialNumber
 OK
$
```

4.1.7 Record the Signature Expiry Time

Update the key-tag file to hold the expiration date of the zone signature.

Zone	**ZSK**				**KSK**				**Exp**
	Tag	Size	Creat	S	Tag	Size	Creat	S	
zone.name	zsktag-cur	512	date	C	ksktag	1024	date	C	date
	zsktag-pub	512	date	P					

4.2 Creating a Signed Delegation in a Child Zone

This section describes the steps required to create a signed delegation in a child zone.

See Figure 4.

4.2.1 Check the Keyset File

Ensure that the keyset file (generated in 4.1.5 or 5.1.5) contains the correct KSK. A visual comparison of the key material in the keyset file with the actual key contents will determine this. Also, ensure that the key-tags in the keyset file correspond to the tags that were recorded for the KSK in in 4.1 or 5.1.

```
$ cat keyset-zonefile [ENTER]
 zone.name                    3600 IN   DNSKEY 257 3 5 (
                                                 ...
                                        ); key id = key-tag
$
```

4.2.2 Securely Transfer the Keyset to the Parent

If any of the zone's KSKs have changed since the last time this file was sent to the parent, then the keyset must also be transferred to the parent. If none of the zone's KSKs have changed, this step may be skipped.

This is not required during a ZSK roll-over. See Sections 3.4, 6.2, 6.3, and 6.4.

Secure communication between the parent and child zone is done out-of-band.

4.2.3 Wait for the Parent to Publish the DS Record

Before proceeding, wait for the parent zone to publish the DS record. This may be found by using the *dig* command to retrieve the zone's DS record. The *aa* flag in the result must be set and the ANSWER section must not be empty.

You may continue if the DS record is the same as the value in the file generated in 4.1.5 or 5.1.5.

```
$ dig @parent-IP-address DS zone.name [ENTER]
```

```
 ; <<>> DiG 9.3.0 <<>> ...
 ...
 ;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0
 ...
 ;;ANSWER SECTION

 zone.name        3600     IN       DS       ...
 ...
$
```

5 Normal Operations for Parent Zones

This section describes those normal DNSSEC operations which are relevant for parent zones. These operations are:

- Signing a Zone that Has Delegations
- Creating a Signed Delegation in a Parent Zone

5.1 Signing a Zone that Has Delegations

A zone needs to be re-signed when any change is made to it. Steps 5.1.1 and 5.1.2 must be followed if the zone has not been previously signed. Steps 5.1.3 - 5.1.8 must be followed when re-signing a zone file that contains delegations.

See Figure 5.

5.1.1 Generate Two ZSKs and One KSK

Follow the steps in Section 3.1 for generating the ZSKs and steps in Section 3.2 for generating the KSK.

Designate one of the two ZSKs as the Current (C) zone signing key and use it to sign the zone data; designate the other as the Published (P) key, which is for future use following a ZSK roll-over. Set the status of each of these keys in the column marked S.

There is only one KSK and this is used to sign the zone apex keyset; mark its status as Current (C). The Published ZSK should be kept more safely[2] than the Current ZSK. The idea is that the Published ZSK can be easily rolled in even if the Current ZSK is compromised (the Current ZSK may have to be kept on-line in some circumstances).

If the KSK has been stored in a more secure location (off-line, more highly protected directory, etc.) then it might be a good idea to store the Published ZSK in the same secure location.

Zone	**ZSK**				**KSK**				**Exp**
	Tag	Size	Creat	S	Tag	Size	Creat	S	
zone.name	zsktag-cur	512	date	C	ksktag	1024	date	C	
	zsktag-pub	512	date	P					

5.1.2 Modify the Zone File

The zone file must be modified to account for the new keys. Add lines to include the KSK and the ZSKs in the zone file. Also, the SOA serial number must be changed so that the zone file's new contents will be recognized.

```
$ vi zonefile [ENTER]
```

[2] It would be a good idea for an operator to apply increased protection mechanisms (physical, file permissions and ownership, network, etc.) to the Published ZSK than are used for the Current ZSK.

```
zone.name IN SOA servername contact (
                                        2005092105 ; Increase current value by 1.
                                                   ; This value may be different
                                                   ; in your zone file.
           ...
 )
...
;; ksk
$INCLUDE "/path/to/Kzone.name.+005+ksktag.key"
;; cur zsk
$INCLUDE "/path/to/Kzone.name.+005+zsktag-cur.key"
;; pub zsk
$INCLUDE "/path/to/Kzone.name.+005+zsktag-pub.key"
...
$
```

5.1.3 Check the Unsigned Zone File for Errors

Ensure that the unsigned zone file was modified correctly.

```
$ named-checkzone zone.name zonefile [ENTER]
 zone zone.name/IN: loaded serial SerialNumber
 OK
$
```

5.1.4 Check Permissions and Ownership on ZSK and KSK Files

The key files must be readable by the *dnssec-signzone* tool and the name server's user id.

5.1.5 Sign the Zone

Use the *dnssec-signzone* command to sign the zone file. This should not be executed unless the keysets from all child zones have been received. (See Section 5.2.)

```
$ dnssec-signzone -g -k /path/to/Kzone.name.+005+ksktag.key
 -o zone.name -e +2592000 -d keyset-dir zonefile
```

```
/path/to/Kzone.name.+005+zsktag-cur.key [ENTER]
zonefile.signed
$
```

Signature generation may take a few minutes to complete, depending on the size of the zone file. If the above operation appears to be unresponsive for an unreasonable length of time, use pseudorandom numbers (using the *-p* option) instead.

```
$ dnssec-signzone -g -k /path/to/Kzone.name.+005+ksktag.key
 -o zone.name -p -e +2592000 -d keyset-dir zonefile
 /path/to/Kzone.name.+005+zsktag-cur.key [ENTER]
 zonefile.signed
$
```

The *-d* option specifies the directory in which the child zone's keyset files have been stored. It there are no keyset files available, run the *dnssec-signzone* command without the *-d keyset-dir* option.

Three files are created by *dnssec-signzone*:

- Signed zone file in zonefile.signed. The *.signed* suffix is appended by default.
- Keyset file in keyset-zone.name. This may have to be sent to the parent zone if this zone is also a child zone; see Section 4.2.
- DS-set file in dsset-zone.name. Used to verify that the correct DS record was generated at the parent; see Section 4.2.

The *dnssec-signzone* command generates signatures for the records that are valid for 30 days (2,592,000 seconds) from the current time. This is offset by -1 hour to account for clock skew between the name server and DNSSEC validators.

5.1.6 Check the Signed Zone File for Errors

Ensure that the signed zone file was modified correctly.

```
$ named-checkzone zone.name zonefile [ENTER]
 zone zone.name/IN: loaded serial SerialNumber
 OK
$
```

5.1.7 Record the Signature Expiry Time

Update the key-tag file to hold the expiration date of the zone signature.

Zone	**ZSK**				**KSK**				**Exp**
	Tag	Size	Creat	S	Tag	Size	Creat	S	
zone.name	zsktag-cur	512	date	C	ksktag	1024	date	C	date
	zsktag-pub	512	date	P					

5.1.8 Confirm DS Record Creation in Signed Zone File

There should be a DS record in the signed zone file (*zone.name.signed*) for every domain name from which a keyset was received.

5.2 Creating a Signed Delegation for a Child Zone

This section describes the steps a parent zone must take in order to create a signed delegation for a child zone.

See Figures 6 and 7.

5.2.1 Ensure that the Child Keysets were Received Over a Secure Channel

Secure communication between the parent and child zone is done out-of-band.

5.2.2 Ensure that Each Received Keyset is for a Delegated Zone

The owner name for the DNSKEY record in the received keyset must correspond to a valid delegation.

```
$ cat keyset-childzonefile [ENTER]
 child.zone.name     3600 IN          DNSKEY 56 3 5 (
                                      ...
                                      ); key id = key-tag
$
```

child.zone.name must exist in the parent zonefile as a valid delegation.

```
$ cat zonefile [ENTER]
 ...
 child.zone.name     NS        server
                     A         ...
 ...
$
```

5.2.3 Store All Child Keysets in a Separate Directory

This is simply to keep things clean.

5.2.4 Sign the Zone

Sign the zone using steps described in Section 5.1.

5.2.5 Reload the Zone

The *rndc* command will reload the name server configuration files and the zone contents. The name server process is assumed to be already running.

```
$ rndc reload zone.name [ENTER]
$
```

6 Emergency Operations for Child Zones

This section describes those emergency DNSSEC operations which are relevant for child zones. These operations are:

- KSK Roll-Over Due to KSK Compromise
- Current ZSK Roll-Over Due to ZSK Compromise
- Published ZSK Roll-Over Due to ZSK Compromise
- Current and Published ZSK Roll-Over Due to ZSK Compromise

6.1 KSK Roll-Over — KSK Compromise

The emergency procedures described for key roll-over use the rationale that injection of valid but false data (which can be generated using the compromised key) is more serious than discontinuity in our ability to validate true data. Thus, during emergency KSK roll-over, there will be a period (up to twice the maximum zone TTL) where it may not be possible to build an "authentication chain" from the zone data to the new KSK.

See Figure 8.

6.1.1 Inform Parent about the KSK Compromise

This communication between parent and child must be done securely using out-of-band mechanisms.

6.1.2 Wait for the Parent to Remove the Zone's DS Record

Before proceeding, wait for the parent zone to remove the DS record. This may be found by using the *dig* command to retrieve the parent's DS record.

The *aa* flag in the result must be set and no answer should be returned for the DS query.

```
$ dig @parent-IP-address DS zone.name [ENTER]
 ; <<>> DiG 9.3.0 <<>> ...
 ...
 ;; flags: qr aa rd; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0
 ...
$
```

6.1.3 Generate a New KSK

Follow the steps in Section 3.2 for generation of a new KSK. Update the key-tag table with the New KSK. Delete the Current KSK. Mark the status of the New KSK as Current (C).

Zone	**ZSK**				**KSK**				**Exp**
	Tag	Size	Creat	S	Tag	Size	Creat	S	
zone.name	zsktag-cur	512	date	C	~~ksktag-cur~~	~~1024~~	~~date~~	~~C~~	date
	zsktag-pub	512	date	P	**ksktag-new**	**1024**	**date**	**C**	

6.1.4 Modify the Zone File

The zone file must be modified to account for the key changes. The Current ZSK must be deleted and the New ZSK must be added. Also, the SOA serial number must be changed so that the zone file's new contents will be recognized.

```
$ vi zonefile [ENTER]
zone.name IN SOA servername contact (
                                        2005092106 ; Increase current value by 1.
                                                   ; This value may be different
                                                   ; in your zone file.
            ...
 )
...
~~;; cur ksk~~
~~$INCLUDE "/path/to/Kzone.name.+005+ksktag-cur.key"~~
;; new ksk
$INCLUDE "/path/to/Kzone.name.+005+ksktag-pub.key"
;; pub zsk
$INCLUDE "/path/to/Kzone.name.+005+zsktag-pub.key"
;; new zsk
$INCLUDE "/path/to/Kzone.name.+005+zsktag-new.key"
...
$
```

6.1.5 Regenerate the ZSKs

The ZSKs can no longer be trusted. Follow the steps in 6.4.1 and 6.4.2 to create a new Current ZSK and Published ZSK.

6.1.6 Sign the Zone with the Current ZSK and the Current KSK

Follow steps 4.1.3 – 4.1.7 if the zone does no delegation. Follow steps 5.1.3 – 5.1.8 if the zone does delegation. The ZSK used in the signing process in Section 4.1.5 or 5.1.5 must be the key that is marked as the Current key (C) in the key-tag table. The KSK is the new key, which has been marked with the status Current.

6.1.7 Perform Steps in Section 4.2 if this Zone is a Secure Delegation from Another Zone

Send the keyset generated from the zone-signing process in 6.1.5 to the parent in order to complete the secure delegation.

6.1.8 Reload the Zone

The *rndc* command will reload the name server configuration files and the zone contents. The name server process is assumed to be already running.

```
$ rndc reload zone.name [ENTER]
$
```

6.1.9 Dispose of the Old KSK

Delete the old KSK's *.private* and *.key* files.

6.2 ZSK Roll-Over — Current ZSK Compromise

If the KSK is also compromised, perform the emergency KSK roll-over first.

As long as there is a valid KSK signature over the ZSK, the KSK can continue to be used to inject false zone data. If both keys are compromised, clients are exposed to attacks[3] on that data until the maximum of the expiration of the KSK's RRSIG (created by the ZSK) and the parent's signature over the DS of that KSK. Short TTLs allow recursive servers to more quickly recover from key-compromise situations, allowing them to get new keys more quickly. Key compromise exposes the secure recursive server to replays of the old key until the signature expires.

The emergency procedures described for key roll-over use the rationale that injection of valid but false data (which can be generated using the compromised key) is more serious than discontinuity in the ability to validate true data. Thus, during emergency ZSK roll-over, there will be a period (up to twice the maximum zone TTL) where the cached zone data may not validate against the new ZSK. Also, the steps below are only useful if the Published and Current keys are kept separate from each other and if the Published ZSK has not also been compromised. If both ZSKs are compromised follow the steps in Section refroll-emergency-zsks. If only the Published key is compromised follow the steps in Section 6.3.

See Figure 9.

6.2.1 Generate a New ZSK

Generate a new ZSK, as described in section 3.1. Update the key-tag table with the new ZSK, and set its status to New (N).

Zone	**ZSK**				**KSK**				**Exp**
	Tag	Size	Creat	S	Tag	Size	Creat	S	
zone.name	zsktag-cur	512	date	C	ksktag	1024	date	C	date
	zsktag-pub	512	date	P					
	zsktag-new	**512**	**date**	**N**					

6.2.2 Modify the Zone File

The zone file must be modified to account for the key changes. The Current ZSK must be deleted and the New ZSK must be added. Also, the SOA serial number must be changed so that the zone file's new contents will be recognized.

[3] These attacks include signatures over false data, replay attacks of the old KSK, and replay attacks of the old DS.

```
$ vi zonefile [ENTER]
zone.name        IN      SOA     servername contact (
                                   2005092102 ; Increase current value by 1.
                                              ; This value may be different
                                              ; in your zone file.
            ...
 )
...
;; ksk
$INCLUDE “/path/to/Kzone.name.+005+ksktag.key”
~~;; cur zsk~~
~~$INCLUDE “/path/to/Kzone.name.+005+zsktag-cur.key”~~
;; pub zsk
$INCLUDE “/path/to/Kzone.name.+005+zsktag-pub.key”
;; new zsk
$INCLUDE “/path/to/Kzone.name.+005+zsktag-new.key”
...
$
```

6.2.3 Sign Zone with the Published ZSK Only

Follow steps 4.1.3 – 4.1.7 if the zone does no delegation. Follow steps 5.1.3 – 5.1.8 if the zone does delegation. The ZSK used in the signing process in Section 4.1.5 or 5.1.5 must be the key marked as the Published key (P) in the key-tag table. The KSK used as input to *dnssec-signzone* does not change, so the keyset does not change and does not have to be re-sent to the parent.

6.2.4 Reload the Zone

The *rndc* command will reload the name server configuration files and the zone contents. The name server process is assumed to be already running.

```
$ rndc reload zone.name [ENTER]
$
```

6.2.5 Update the Key-Tags Table

Update the key-tags table to reflect the changed key status. Delete the old Current ZSK. Change the status of the Published ZSK to Current. Change the status of the New ZSK to Published.

Zone	**ZSK**				**KSK**				**Exp**
	Tag	Size	Creat	S	Tag	Size	Creat	S	
zone.name	~~**zsktag-cur**~~	~~**512**~~	~~**date**~~	~~C~~	ksktag	1024	date	C	date
	zsktag-pub	512	date	~~**P**~~ **C**					
	zsktag-new	512	date	~~**N**~~ **P**					

6.2.6 Dispose of the Old Zone Key

Delete the old ZSK's *.private* and *.key* files.

6.3 ZSK Roll-Over — Published ZSK Compromise

If the KSK is also compromised, perform the emergency KSK roll-over first.

As long as there is a valid KSK signature over the ZSK, the KSK can continue to be used to inject false zone data. If both keys are compromised, clients are exposed to attacks[4] on that data until the maximum of the expiration of the KSK's RRSIG (created by the ZSK) and the parent's signature over the DS of that KSK. Short TTLs allow recursive servers to more quickly recover from key-compromise situations, allowing them to get new keys more quickly. Key compromise exposes the secure recursive server to replays of the old key until the signature expires.

The emergency procedures described for key roll-over uses that rationale that injection of valid but false data (which can be generated using the compromised key) is more serious than discontinuity in the ability to validate true data. Thus, during emergency ZSK roll-over, there will be a period (up to twice the maximum zone TTL) where the cached zone data may not validate against the new ZSK.

See Figure 9.

6.3.1 Generate a New ZSK

Generate a new ZSK, as described in section 3.1. Update the key-tag table with the new ZSK, and set its status to New (N).

Zone	**ZSK**				**KSK**				**Exp**
	Tag	Size	Creat	S	Tag	Size	Creat	S	
zone.name	zsktag-cur	512	date	C	ksktag	1024	date	C	date
	zsktag-pub	512	date	P					
	zsktag-new	**512**	**date**	**N**					

6.3.2 Modify the Zone File

The zone file must be modified to account for the key changes. The Current ZSK must be deleted and the New ZSK must be added. Also, the SOA serial number must be changed so that the zone file's new contents will be recognized.

```
$ vi zonefile [ENTER]
zone.name            IN        SOA       servername contact (
```

[4] These attacks include signatures over false data, replay attacks of the old KSK, and replay attacks of the old DS.

```
                                        2005092107 ; Increase current value by 1.
                                                   ; This value may be different
                                                   ; in your zone file.
                 ...
 )
...
;; ksk
$INCLUDE "/path/to/Kzone.name.+005+ksktag.key"
;; cur zsk
$INCLUDE "/path/to/Kzone.name.+005+zsktag-cur.key"
~~;; pub zsk~~
~~$INCLUDE "/path/to/Kzone.name.+005+zsktag-pub.key"~~
;; new zsk
$INCLUDE "/path/to/Kzone.name.+005+zsktag-new.key"
...
$
```

6.3.3 Sign the Zone with the KSK and Current ZSK

Follow the steps 4.1.3 – 4.1.7 if the zone does no delegation. Follow the steps 5.1.3 – 5.1.8 if the zone does delegation. The ZSK used in the signing process in Section 4.1.5 or 5.1.5 must be the key that is marked as the Current key (C) in the key-tag table (this was the older Published key.) The KSK used as input to *dnssec-signzone* does not change, so the keyset does not change and does not have to be re-sent to the parent.

6.3.4 Reload the Zone

The *rndc* command will reload the name server configuration files and the zone contents. The name server process is assumed to be already running.

```
$ rndc reload zone.name [ENTER]
$
```

6.3.5 Update the Key-Tags Table

Update the key-tag table to reflect the changed key status. Delete the old Published ZSK. Change the status of the New ZSK to Published.

Zone	**ZSK**				**KSK**				**Exp**
	Tag	Size	Creat	S	Tag	Size	Creat	S	
zone.name	zsktag-cur	512	date	C					date
	~~**zsktag-pub**~~	~~**512**~~	~~**date**~~	~~**P**~~	ksktag	1024	date	C	
	zsktag-new	512	date	~~**N**~~ **P**					

6.3.6 Dispose of the Old Zone Key

Delete the old ZSK's *.private* and *.key* files.

6.4 ZSK Roll-Over — Current and Published ZSK Compromise

If the KSK is also compromised, perform the emergency KSK roll-over first.

The emergency procedures described for key roll-over use the rationale that injection of valid but false data (which can be generated using the compromised key) is more serious than discontinuity in our ability to validate true data. Thus, during emergency ZSK roll-over, there will be a period (up to twice the maximum zone TTL) where the cached zone data may not validate against the new ZSK.

See Figure 9.

6.4.1 Generate New Current and Published ZSKs

Follow the steps in Section 3.1 in order to generate two ZSKs. Update the key-tag table with the new ZSKs; replace the existing set of ZSKs with the new values.

Zone	**ZSK**				**KSK**				**Exp**
	Tag	Size	Creat	S	Tag	Size	Creat	S	
zone.name	~~**zsktag-cur**~~	~~**512**~~	~~**date**~~	~~C~~	ksktag	1024	date	C	date
	~~**zsktag-pub**~~	~~**512**~~	~~**date**~~	~~P~~					
	zsktag-cur	**512**	**date**	C					
	zsktag-pub	**512**	**date**	**P**					

6.4.2 Modify the Zone File

The zone file must be modified to account for the key changes. The Current ZSK must be deleted and the New ZSK must be added. Also, the SOA serial number must be changed so that the zone file's new contents will be recognized.

```
$ vi zonefile [ENTER]
zone.name        IN      SOA     servername contact (
                                  2005092108 ; Increase current value by 1.
                                             ; This value may be different
                                             ; in your zone file.
            ...
 )
...
;; ksk
$INCLUDE "/path/to/Kzone.name.+005+ksktag.key"
~~;; cur zsk~~
```

```
~~$INCLUDE "/path/to/Kzone.name.+005+zsktag-cur.key"~~
~~;; pub zsk~~
~~$INCLUDE "/path/to/Kzone.name.+005+zsktag-pub.key"~~
;; cur zsk
$INCLUDE "/path/to/Kzone.name.+005+zsktag-cur.key"
;; pub zsk
$INCLUDE "/path/to/Kzone.name.+005+zsktag-pub.key"
...
$
```

6.4.3 Sign the Zone with the KSK and Current ZSK

Follow the steps 4.1.3 – 4.1.7 if the zone does no delegation. Follow the steps 5.1.3 – 5.1.8 if the zone does delegation. The ZSK used in the signing process in Section 4.1.5 or 5.1.5 must be the key that is marked as the Current key (C) in the key-tag table (this was the older Published key.) The KSK used as input to *dnssec-signzone* does not change, so the keyset does not change and does not have to be re-sent to the parent.

6.4.4 Reload the Zone

The *rndc* command will reload the name server configuration files and the zone contents. The name server process is assumed to be already running.

```
$ rndc reload zone.name [ENTER]
$
```

6.4.5 Dispose of the Old Zone Key

Delete the old ZSK's *.private* and *.key* files.

7 Emergency Operations for Parent Zones

This section describes those emergency DNSSEC operations which are relevant for parent zones. These operations are:

- KSK Roll-Over Due to Child KSK Compromise

7.1 KSK Roll-Over — Parent Action During KSK Compromise

During a KSK compromise the child zone is no longer secure. This change in status is performed by deleting the child's DS record from the parent zone.

See Figure 9.

7.1.1 Ensure that the KSK Compromise Notification Came Over a Secure Channel

Authentication and communication between parent and child occurs out-of-band.

7.1.2 Delete the Child's Keyset File at the Parent

The DS record for the child should not be created. This can be achieved by not having the child's keyset file available to the *dnssec-signzone* process.

7.1.3 Increase the Zone SOA Number

Although there is no change made to the unsigned zone file, the DS record for the child will no longer be present in the signed zone file. This amounts to an implicit change. Thus, the zone serial number must be incremented.

```
$ vi zonefile [ENTER]
zone.name IN SOA servername contact (
                                        2005092109 ; Increase current value by 1.
                                                   ; This value may be different
                                                   ; in your zone file.
          ...
 )
```

7.1.4 Reload the Zone

The *rndc* command will reload the name server configuration files and the zone contents. The name server process is assumed to be already running.

```
$ rndc reload zone.name [ENTER]
```

```
$
```

A Operation Checklists

A.1 Normal Operations for All Zones

A.1.1 Zone-Signing Key (ZSK) Generation

□ Generate the Key

□ Update the Key-Tags Table

□ Store Separately the Private Keys for ZSKs and KSKs

A.1.2 Key-Signing Key (KSK) Generation

□ Generate the Key

□ Update the Key-Tags Table

□ Store Separately the Private Keys for ZSKs and KSKs

A.1.3 Configuring and Serving a Signed Zone

□ Add the Signed Zone to the Name Server Configuration File

□ Enable DNSSEC

□ Check the Name Server Configuration File for Errors

□ Reload the Zone

□ Check that the Zone Loaded Properly

A.1.4 Current ZSK Roll-Over

□ Ensure that Sufficient Time has Elapsed Since the Last Roll-Over

□ Sign Zone with the KSK and Published ZSK

□ Reload the Zone

□ Wait for Old Zone Data to Expire from Caches

□ Generate a New ZSK

□ Modify the Zone File

□ Update the Key-Tags Table

□ Sign the Zone with the KSK and Current ZSK

□ Reload the Zone

□ Dispose of the Old Zone Key

A.1.5 KSK Roll-Over

□ Ensure that Sufficient Time has Elapsed Since the Last Roll-Over

□ Generate a New KSK

□ Modify the Zone File

□ Re-Sign the Zone DNSKEY RRset with the Current and Published KSK

□ Reload the Zone

□ Wait for Old DNSKEY RRset to Expire from Caches

□ Modify the Zone File

□ Re-Sign the Zone DNSKEY RRset with the Current and Published KSK

□ Update the Key-Tags Table with the Latest KSK

□ Perform Steps in Section 4.2 if this Zone is a Secure Delegation from Another Zone

□ Reload the Zone

□ Dispose of the Old KSK

A.2 Normal Operations for Child Zones

A.2.1 Signing a Zone with No Delegations

□ Generate Two ZSKs and One KSK

□ Modify the Zone File

□ Check the Unsigned Zone File for Errors

□ Check Permissions and Ownership on ZSK and KSK Files

□ Sign the Zone

□ Check the Signed Zone File for Errors

□ Record the Signature Expiry Time

A.2.2 Creating a Signed Delegation in a Child Zone

□ Check the Keyset File

□ Securely Transfer the Keyset to the Parent

□ Wait for the Parent to Publish the DS Record

A.3 Normal Operations for Parent Zones

A.3.1 Signing a Zone that Has Delegations

□ Generate Two ZSKs and One KSK

□ Modify the Zone File

□ Check the Unsigned Zone File for Errors

□ Check Permissions and Ownership on ZSK and KSK Files

□ Sign the Zone

□ Check the Signed Zone File for Errors

□ Record the Signature Expiry Time

□ Confirm DS Record Creation in Signed Zone File

A.3.2 Creating a Signed Delegation for a Child Zone

□ Ensure that the Child Keysets were Received Over a Secure Channel

□ Ensure that Each Received Keyset is for a Delegated Zone

□ Store All Child Keysets in a Separate Directory

□ Sign the Zone

□ Reload the Zone

A.4 Emergency Operations for Child Zones

A.4.1 KSK Roll-Over – KSK Compromise

□ Inform Parent about the KSK Compromise

□ Wait for the Parent to Remove the Zone's DS Record

□ Generate a New KSK

□ Modify the Zone File

□ Regenerate the ZSKs

□ Sign the Zone with the Current ZSK and the Current KSK

□ Perform Steps in Section 4.2 if this Zone is a Secure Delegation from Another Zone

□ Reload the Zone

□ Dispose of the Old KSK

A.4.2 ZSK Roll-Over – Current ZSK Compromise

□ Generate a New ZSK

□ Modify the Zone File

□ Sign Zone with the Published ZSK Only

□ Reload the Zone

□ Update the Key-Tags Table

□ Dispose of the Old Zone Key

A.4.3 ZSK Roll-Over – Published ZSK Compromise

□ Generate a New ZSK

□ Modify the Zone File

□ Sign the Zone with the KSK and Current ZSK

□ Reload the Zone

□ Update the Key-Tags Table

□ Dispose of the Old Zone Key

A.4.4 ZSK Roll-Over – Current and Published ZSK Compromise

□ Generate New Current and Published ZSKs

□ Modify the Zone File

□ Sign the Zone with the KSK and Current ZSK

□ Reload the Zone

□ Dispose of the Old Zone Key

A.5 Emergency Operations for Parent Zones

A.5.1 KSK Roll-Over – Parent Action During KSK Compromise

□ Ensure that the KSK Compromise Notification Came Over a Secure Channel

□ Delete the Child's Keyset File at the Parent

□ Increase the Zone SOA Number

□ Reload the Zone

B Key-Tag Table Template

Zone	**ZSK**				**KSK**				**Exp**
	Tag	Size	Creat	S	Tag	Size	Creat	S	

C Illustrations

The following illustrations may clarify actions taken for some of the operations described in this guide.

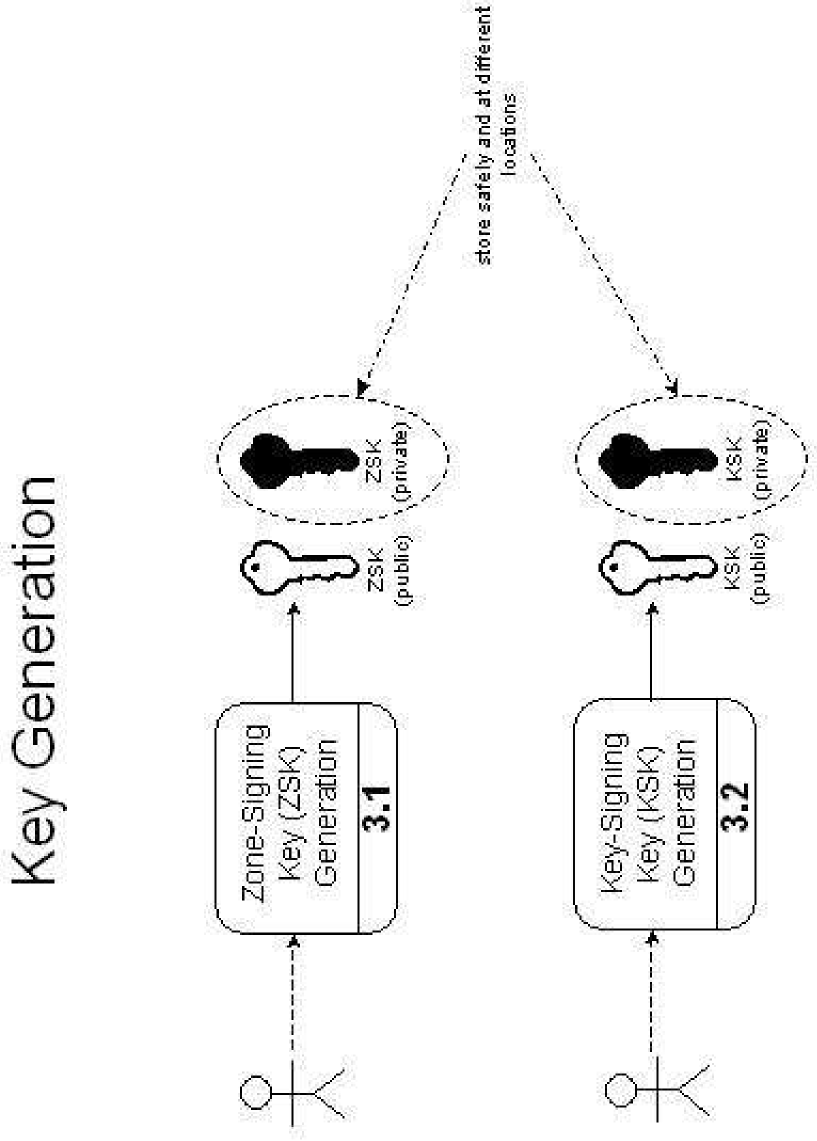
Key Generation
Zone-Signing Key (ZSK) Generation
3.1
Key-Signing Key (KSK) Generation
3.2
ZSK (public)
ZSK (private)
KSK (public)
KSK (private)
store safely and at different locations

Figure 1: Key Generation

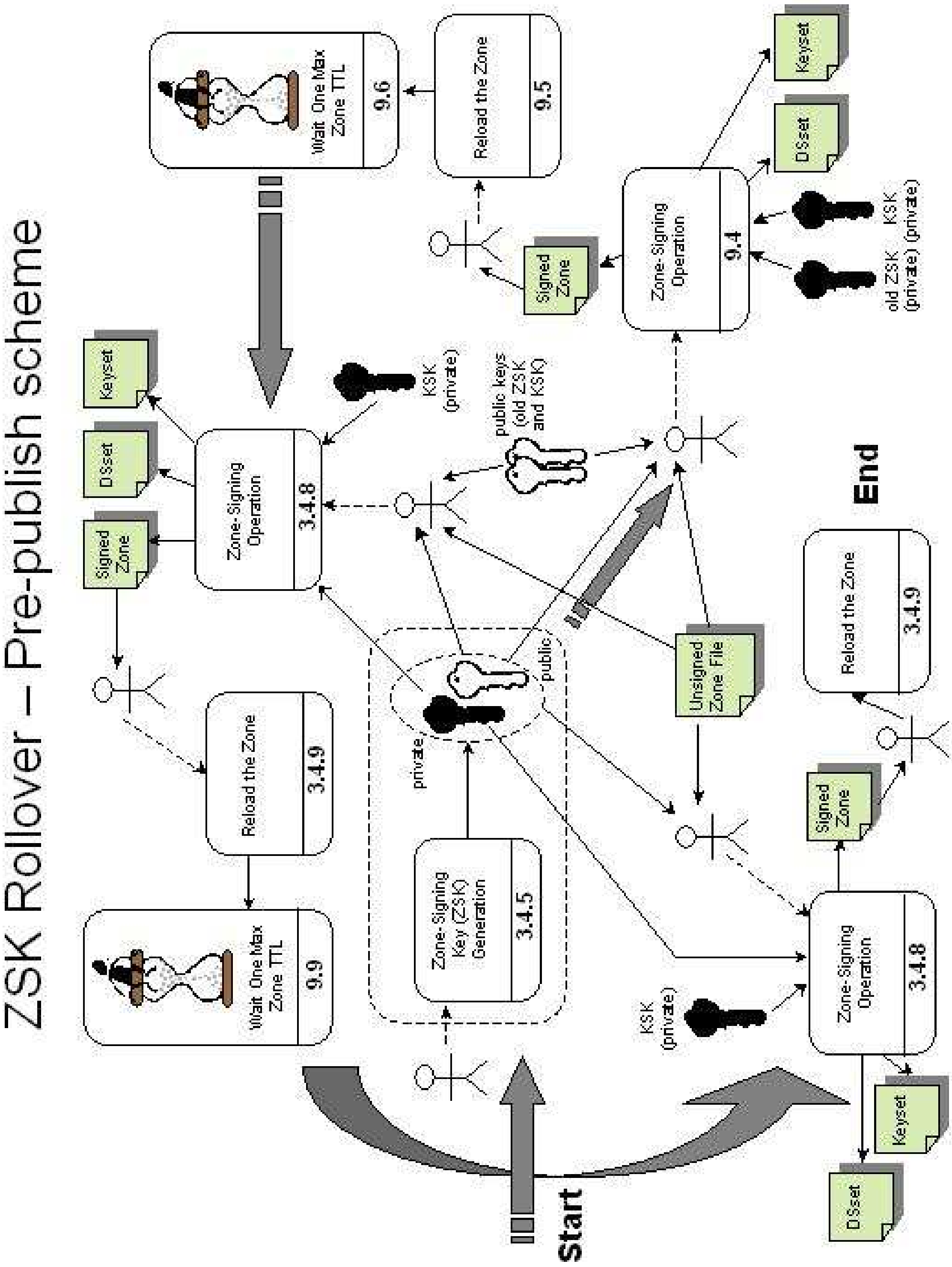

Figure 2: ZSK Roll-Over

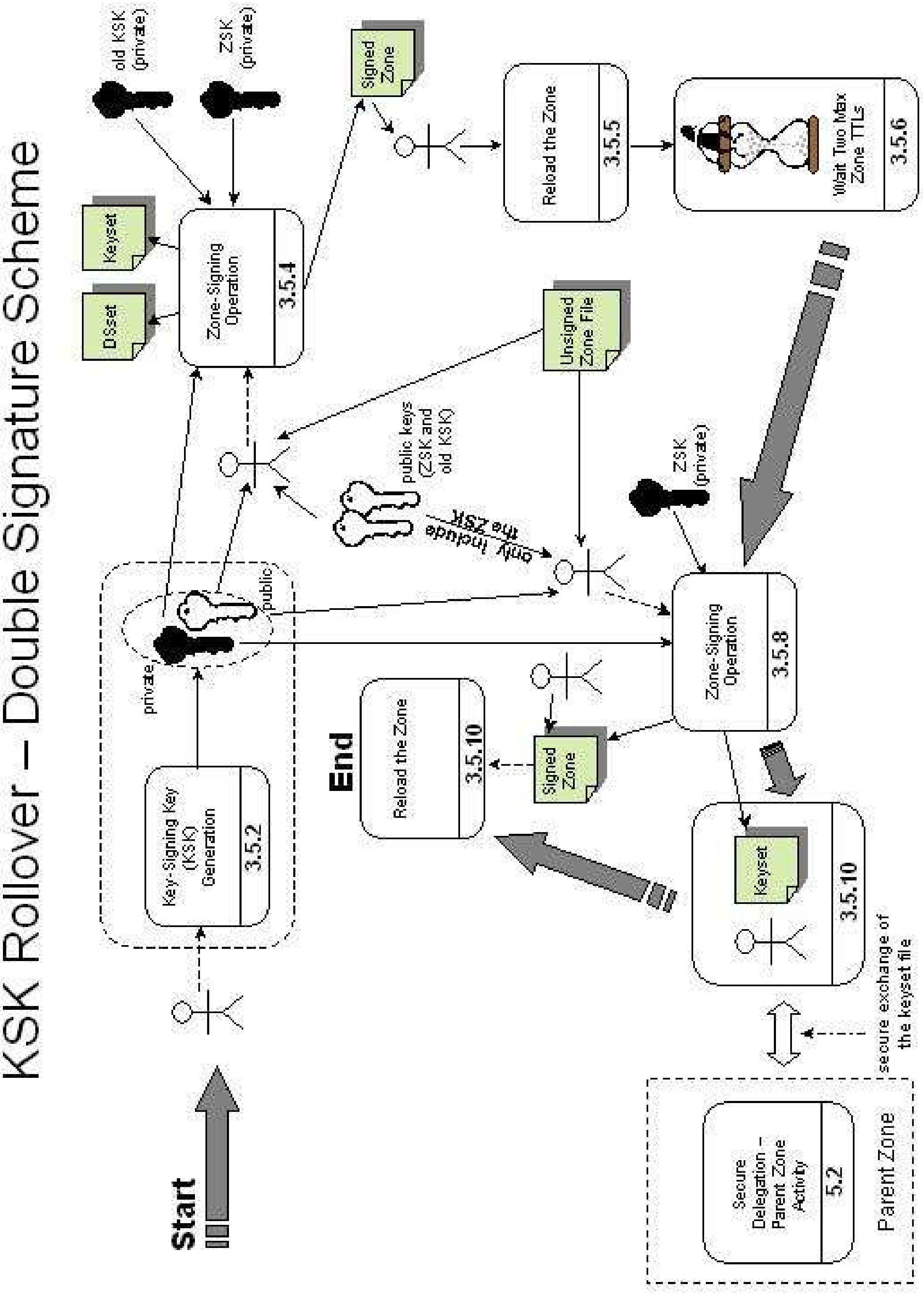

Figure 3: KSK Roll-Over

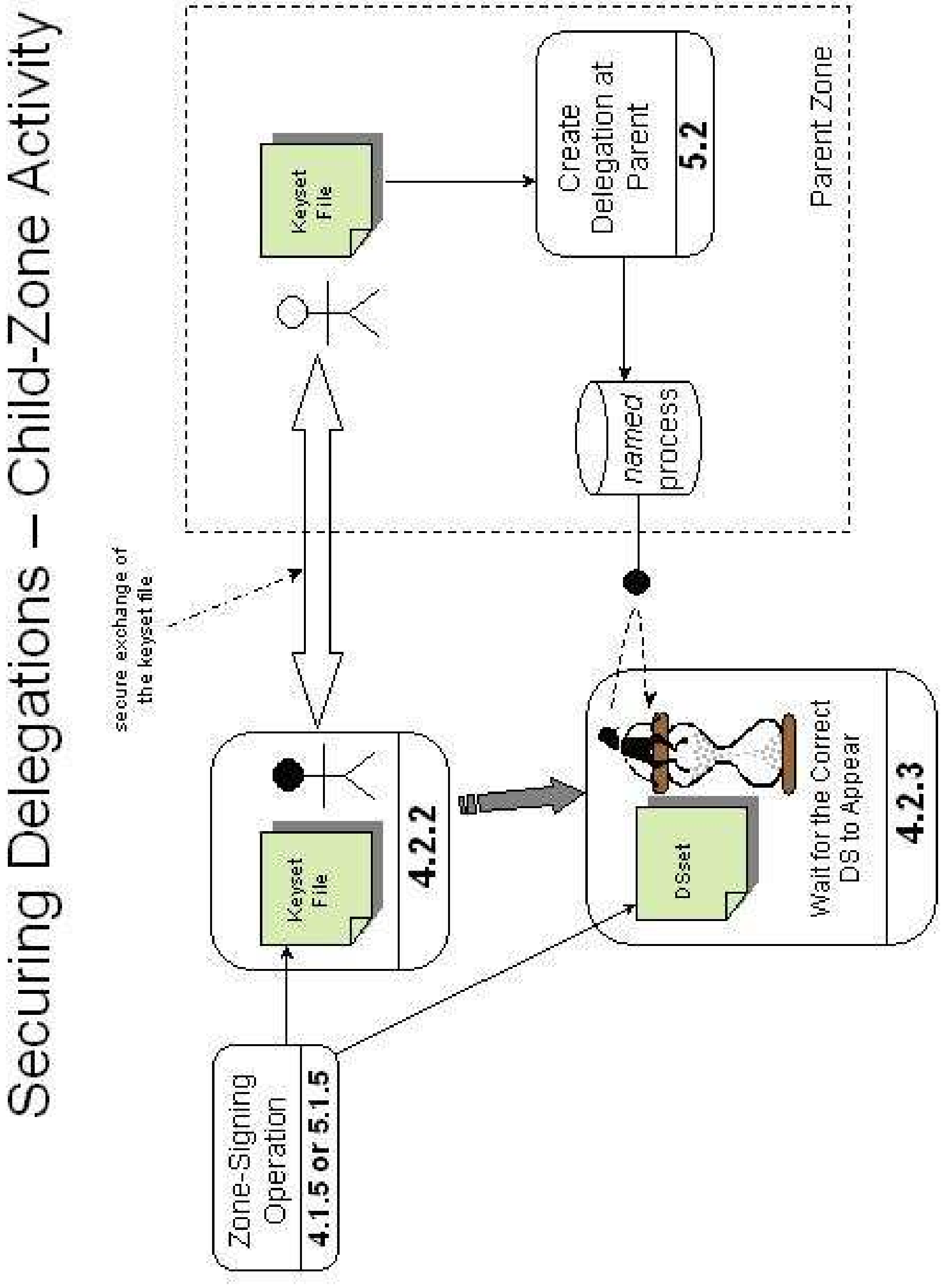
Securing Delegations – Child-Zone Activity
Zone-Signing Operation
4.1.5 or 5.1.5
Keyset File
4.2.2
secure exchange of the keyset file
DSset
Wait for the Correct DS to Appear
4.2.3
Keyset File
Create Delegation at Parent
5.2
named process
Parent Zone

Figure 4: Securing Delegations – Child-Zone Activity

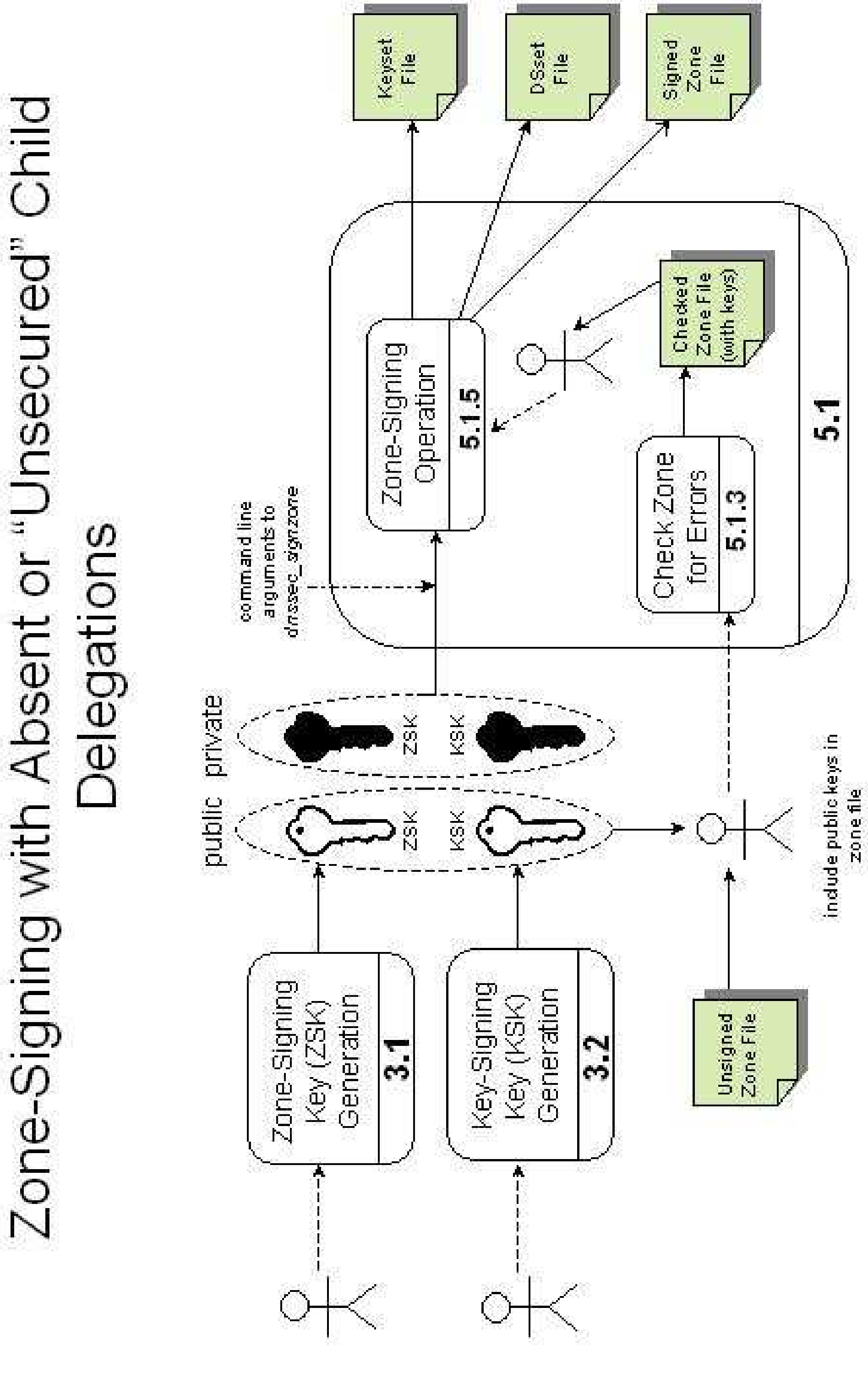
Zone-Signing with Absent or “Unsecured” Child Delegations
Zone-Signing Key (ZSK) Generation
3.1
Key-Signing Key (KSK) Generation
3.2
public
private
ZSK
KSK
ZSK
KSK
Unsigned Zone File
include public keys in zone file
command line arguments to dnssec_signzone
Zone-Signing Operation
5.1.5
Check Zone for Errors
5.1.3
Checked Zone File (with keys)
5.1
Keyset File
DSset File
Signed Zone File

Figure 5: Zone-Signing with Unsecured Child Delegations

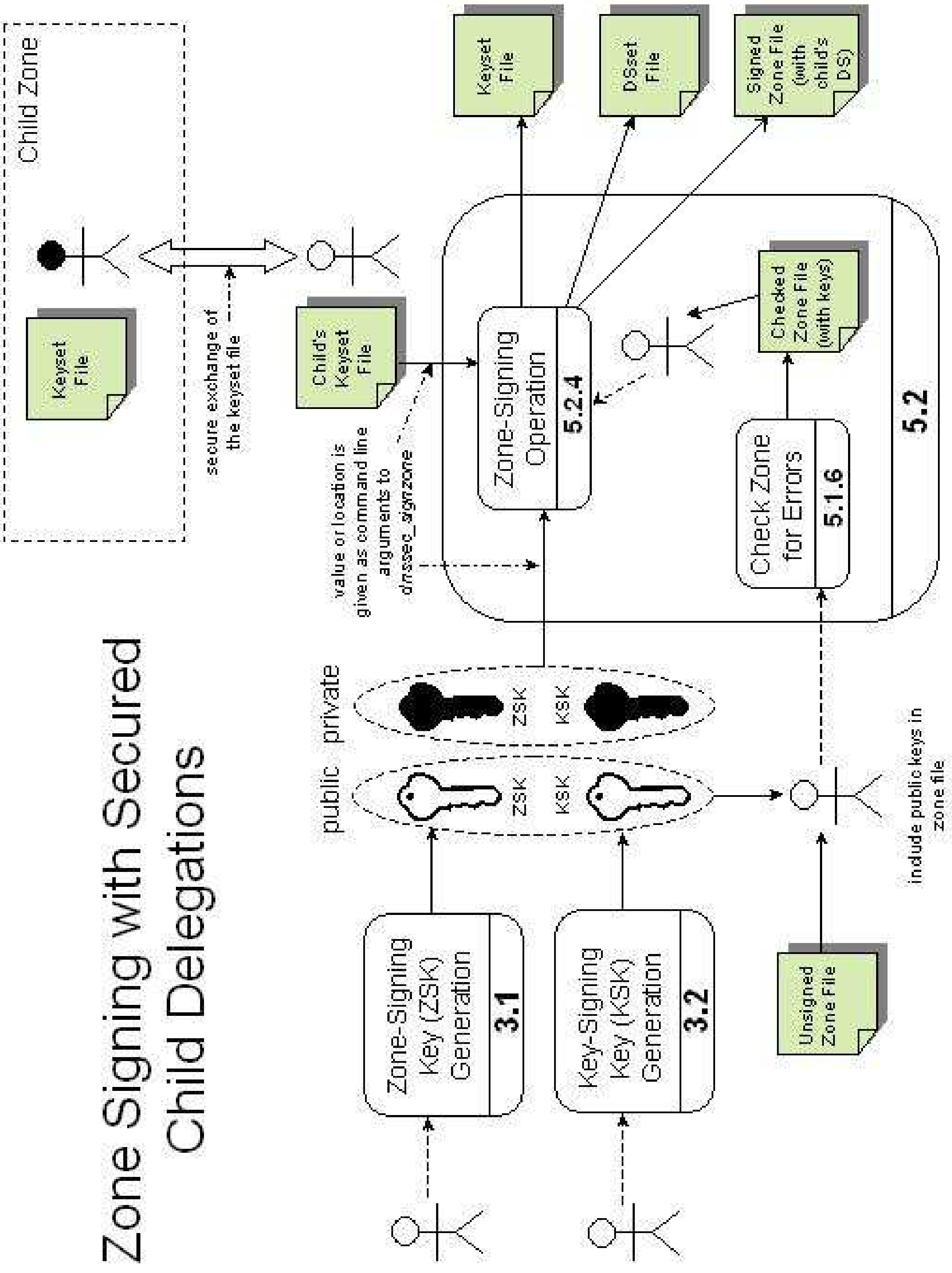
Zone Signing with Secured Child Delegations
Zone-Signing Key (ZSK) Generation
3.1
Key-Signing Key (KSK) Generation
3.2
Unsigned Zone File
public
private
ZSK
KSK
ZSK
KSK
include public keys in zone file
value or location is given as command line arguments to dnssec_signzone
Zone-Signing Operation
5.2.4
Check Zone for Errors
5.1.6
Checked Zone File (with keys)
5.2
Child's Keyset File
Child Zone
Keyset File
secure exchange of the keyset file
Keyset File
DSset File
Signed Zone File (with child's DS)

Figure 6: Zone-Signing with Secured Child Delegations

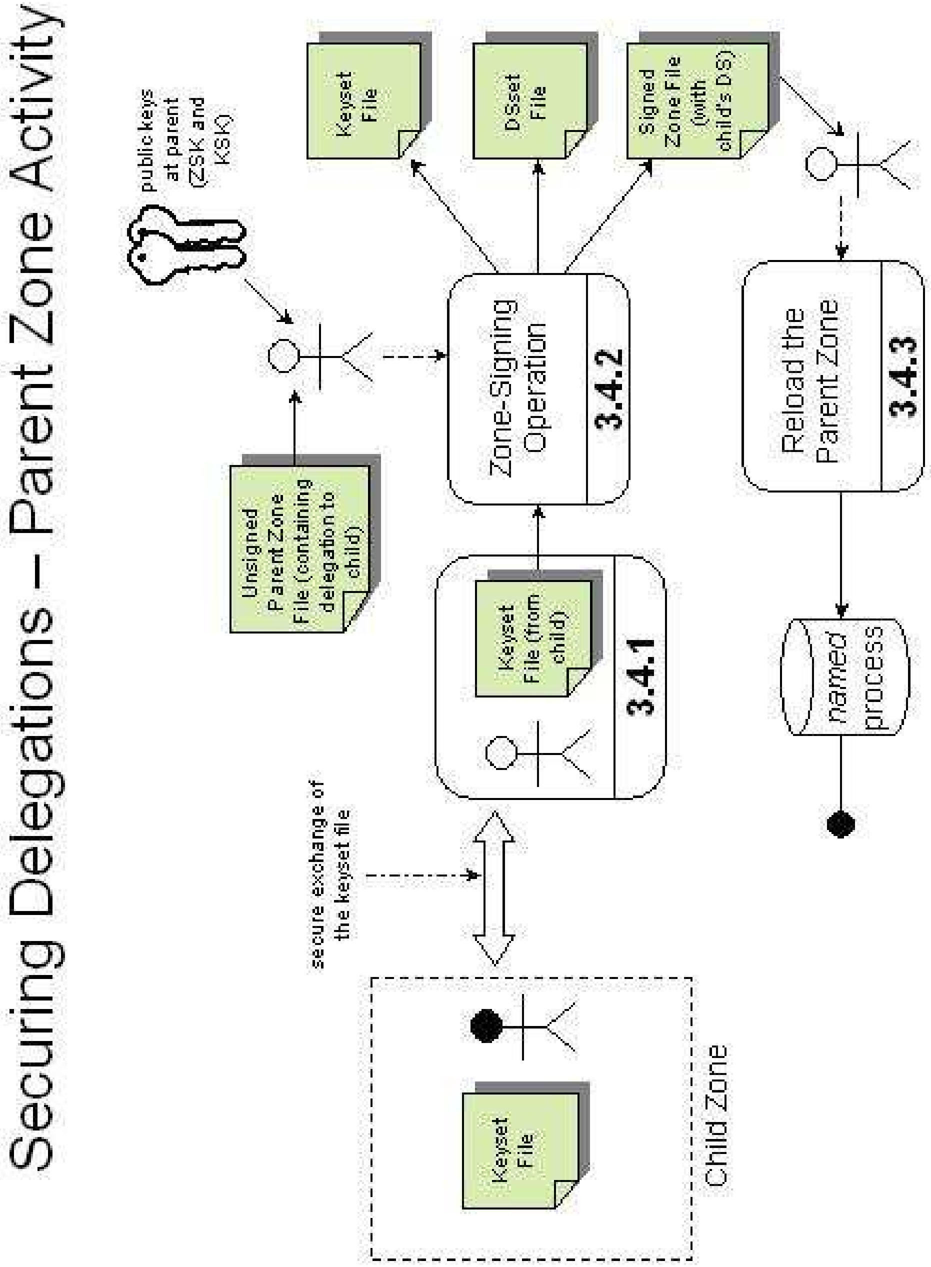
Securing Delegations – Parent Zone Activity
public keys at parent (ZSK and KSK)
Keyset File
DSset File
Signed Zone File (with child's DS)
Unsigned Parent Zone File (containing delegation to child)
Zone-Signing Operation
3.4.2
Keyset File (from child)
3.4.1
Reload the Parent Zone
3.4.3
named process
secure exchange of the keyset file
Keyset File
Child Zone

Figure 7: Securing Delegations – Parent-Zone Activity

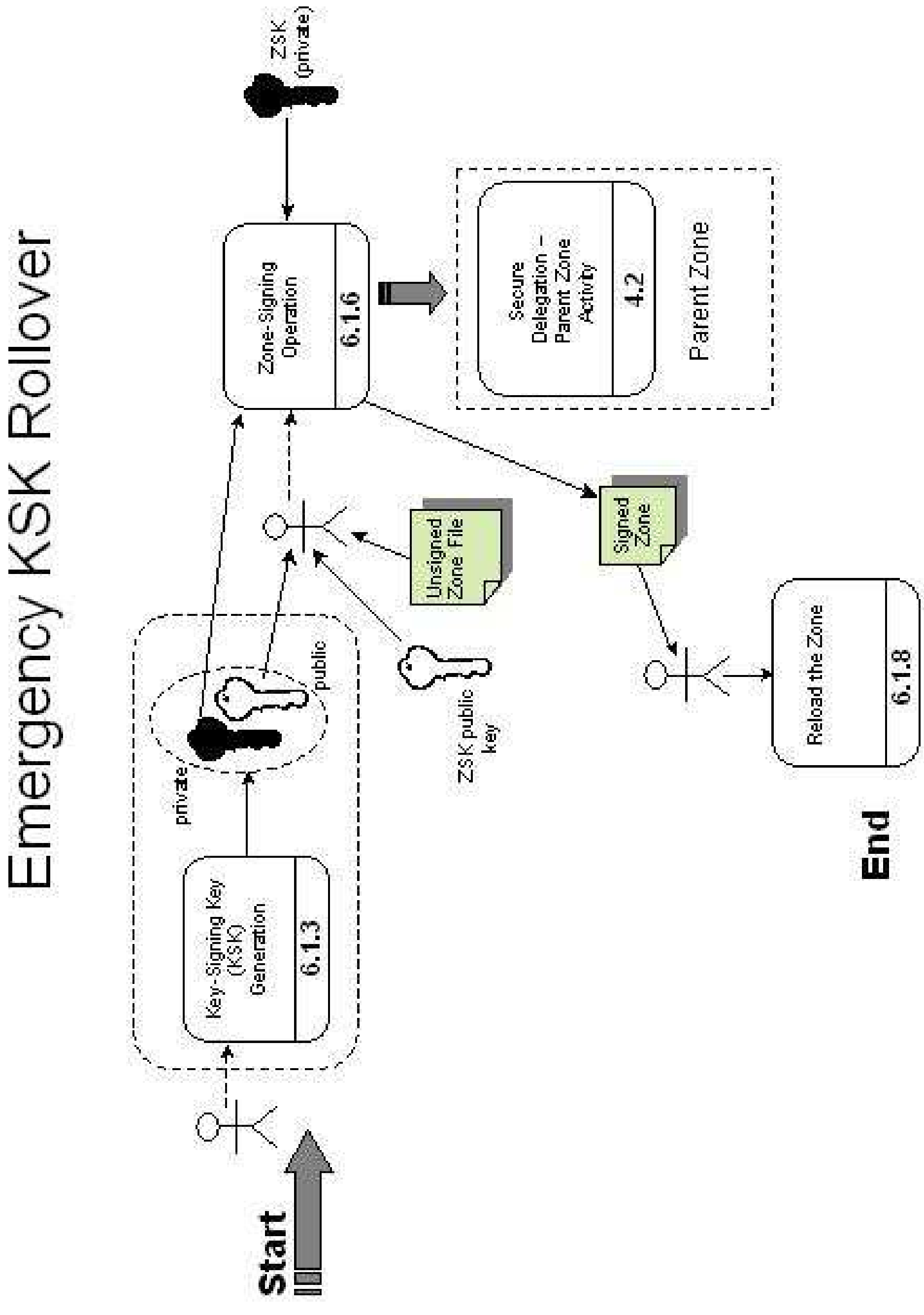

Figure 8: Emergency KSK Roll-Over

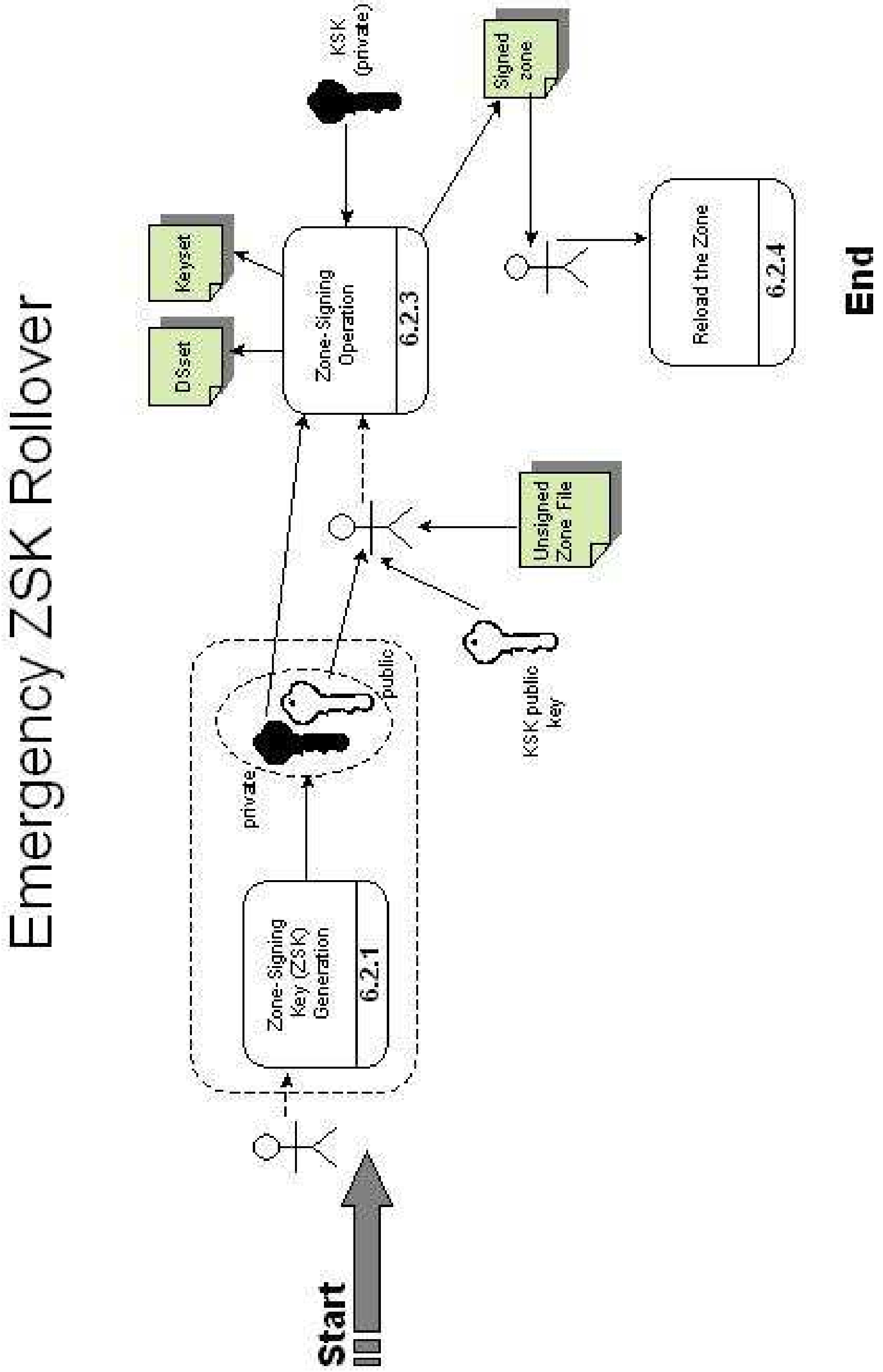

Figure 9: Emergency ZSK Roll-Over

D References

[1] O. Kolkman, R. Gieben, "DNSSEC Operational Practices", *draft-ietf-dnsop-dnssec--operational-practices-03* (work in progress), Dec. 23, 2004.