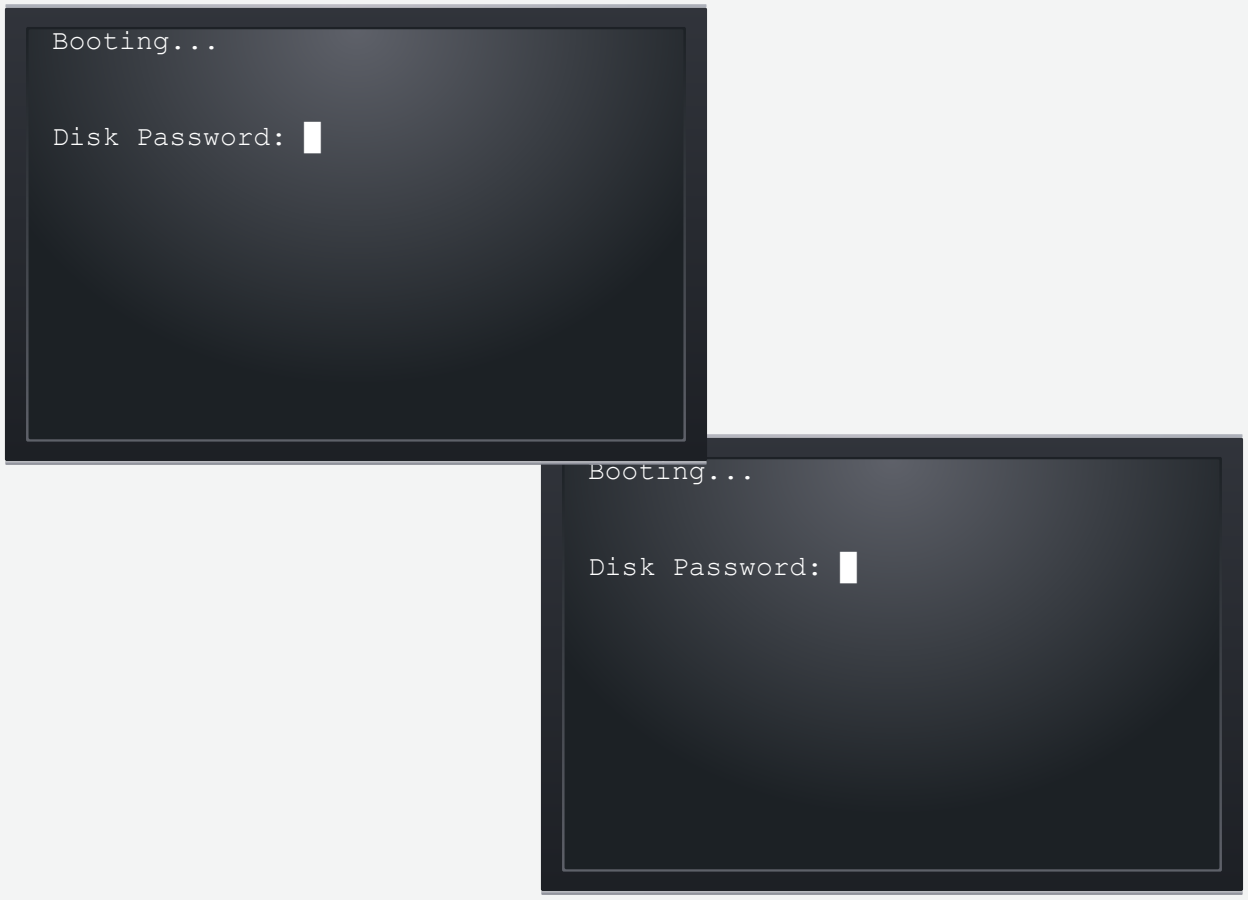
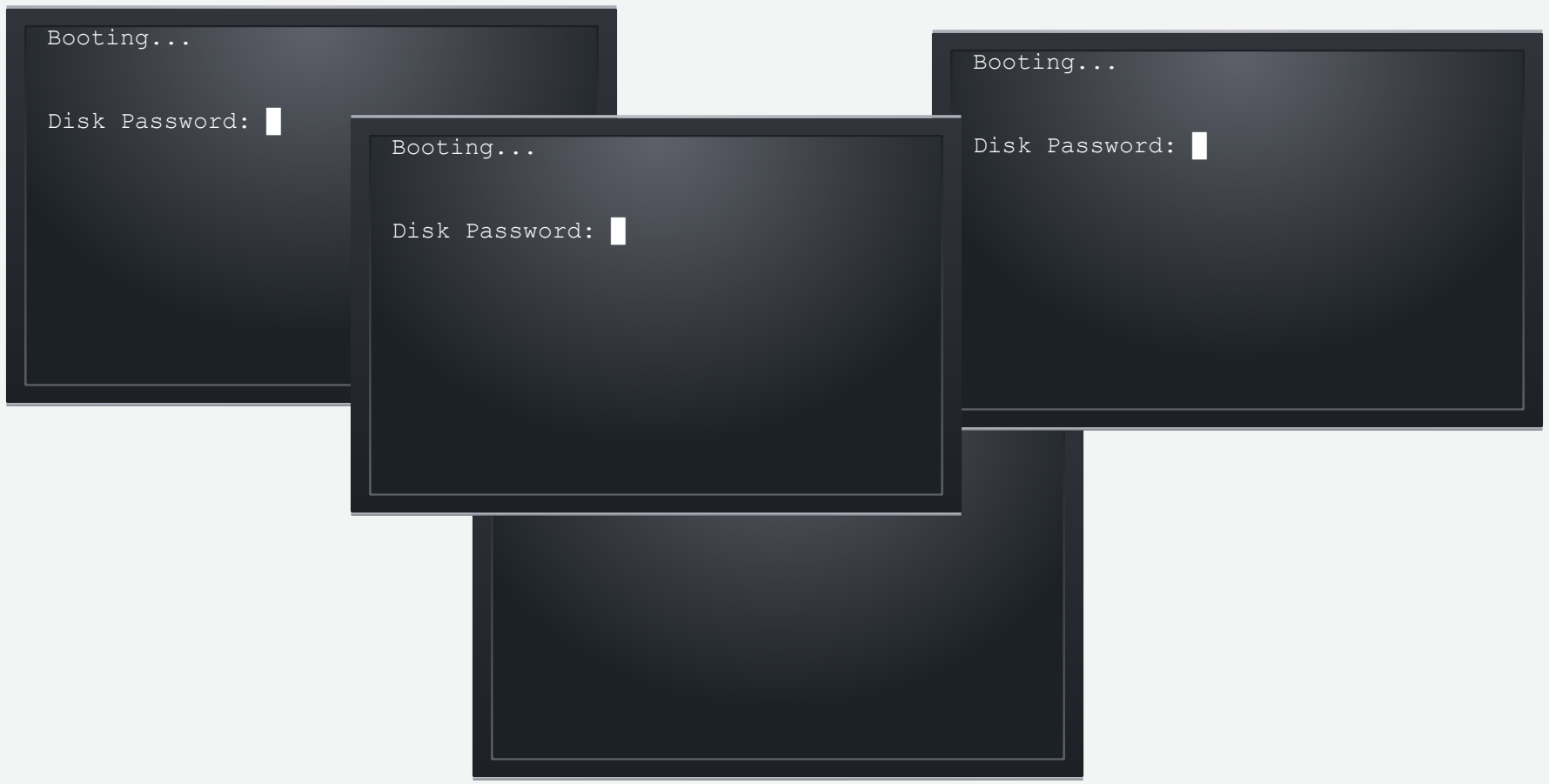
NETWORK-BOUND DISK ENCRYPTION

NYRHUG Monthly Meeting - February 2020

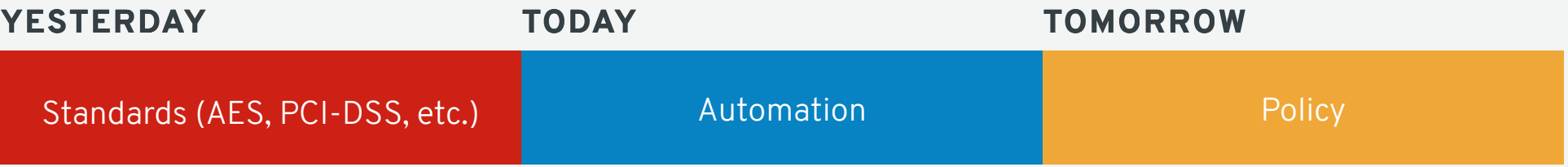
Patrick Ladd
Technical Account Manager - FSI
pladd@redhat.com
http://people.redhat.com/pladd


Booting...
Disk Password:

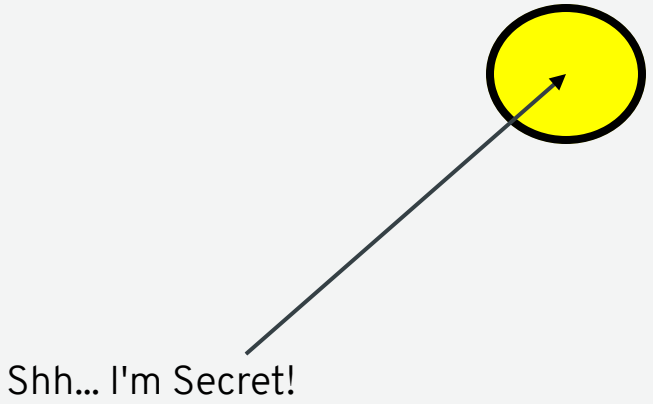

Booting...
Disk Password:
Booting...
Disk Password:

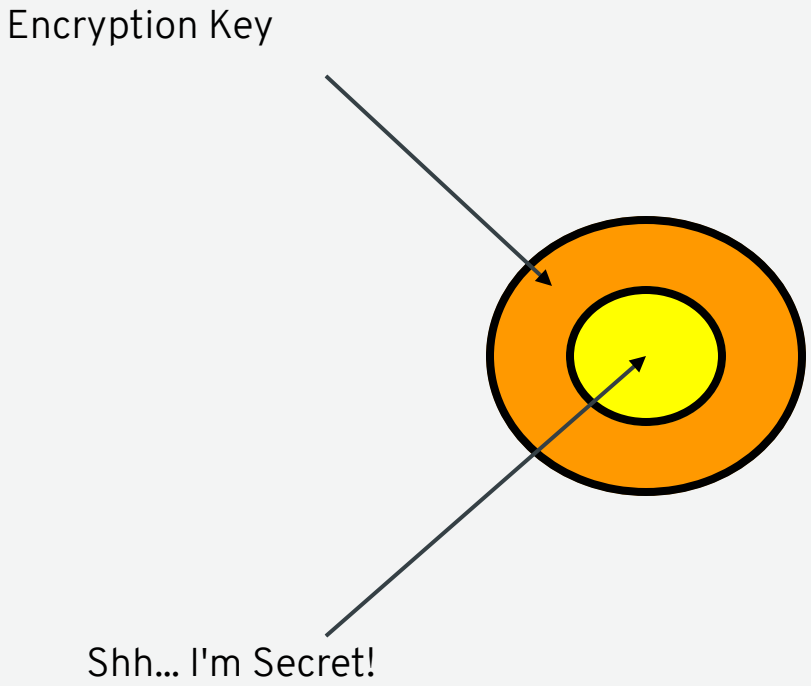

Booting...
Disk Password:
Booting...
Disk Password:
Booting...
Disk Password:

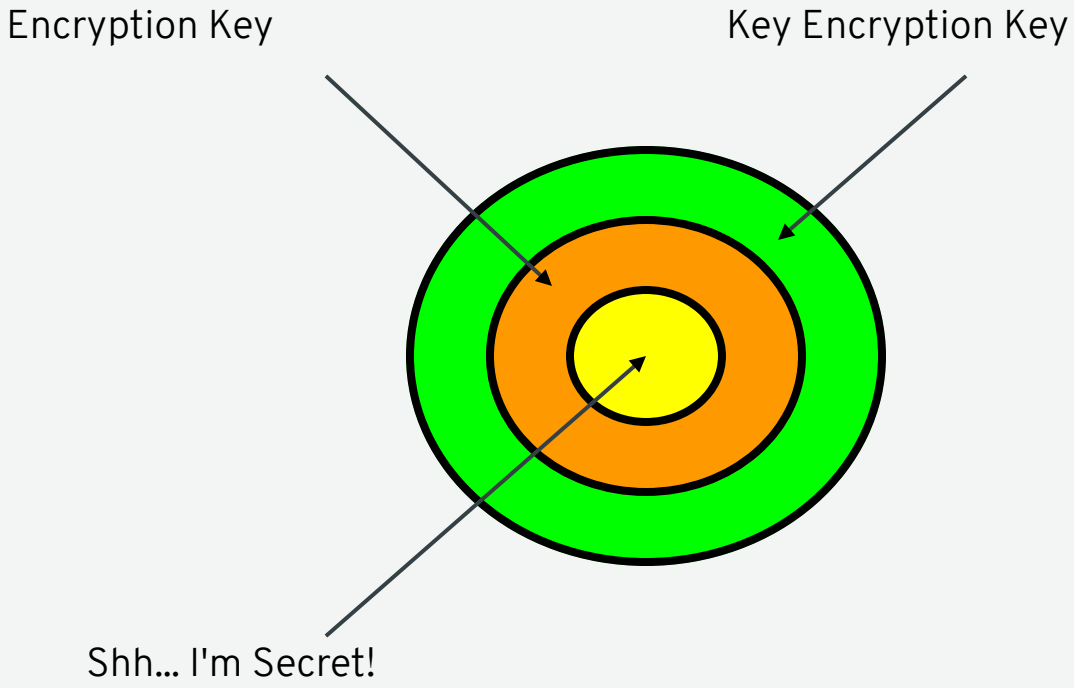

Booting...
Disk Password:
Booting...
Disk Password:
Booting...
Disk Password:
Booting...
Disk Password:
Booting...
Disk Password:
Booting...
Disk Password:

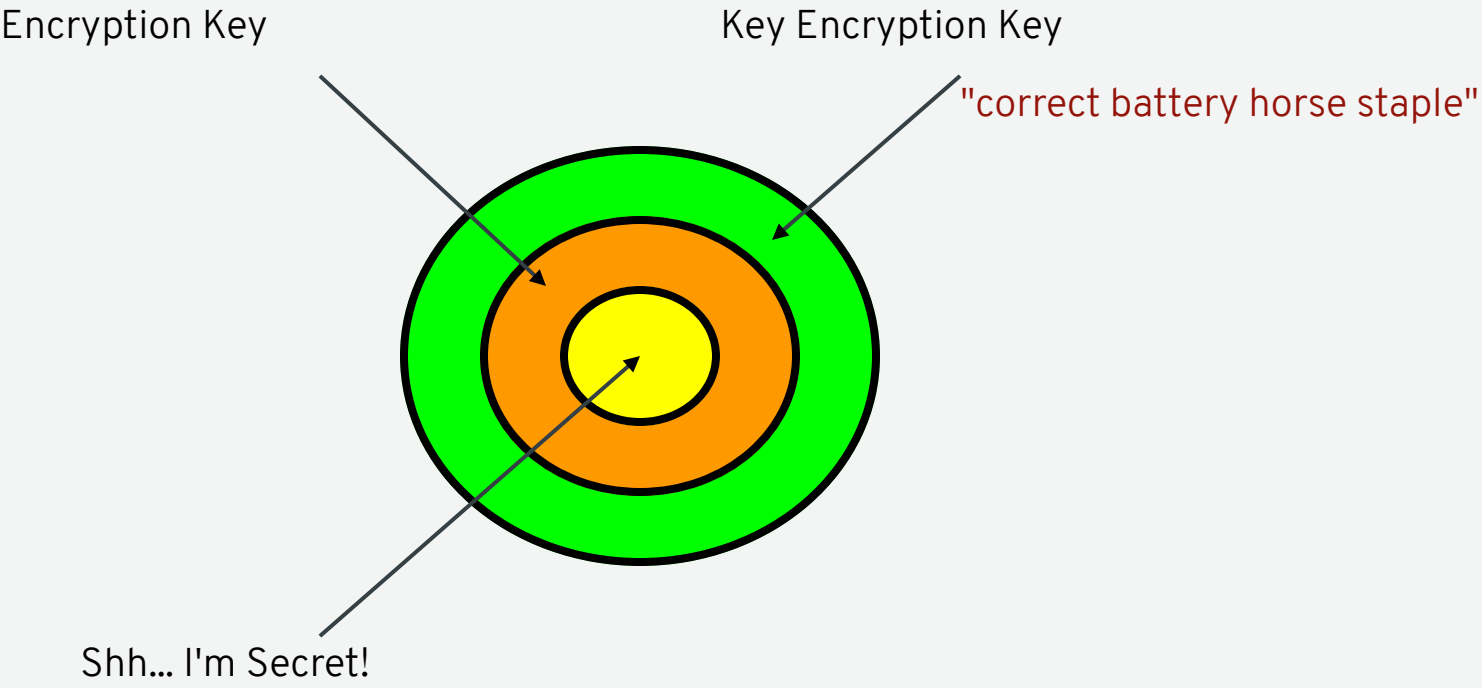

YESTERDAY
TODAY
TOMORROW
Standards (AES, PCI-DSS, etc.)
Automation
Policy

HOW DO WE AUTOMATE?

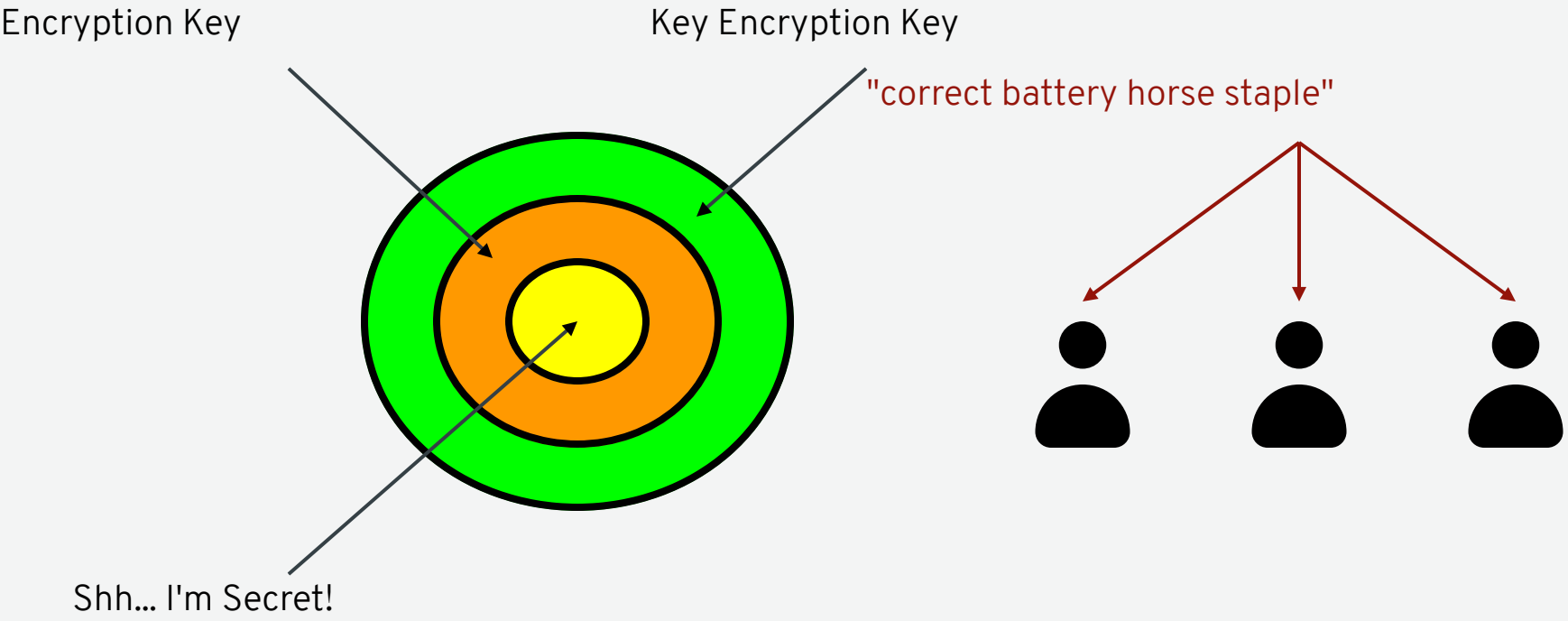

Shh... I'm Secret!


Encryption Key
Shh... I'm Secret!

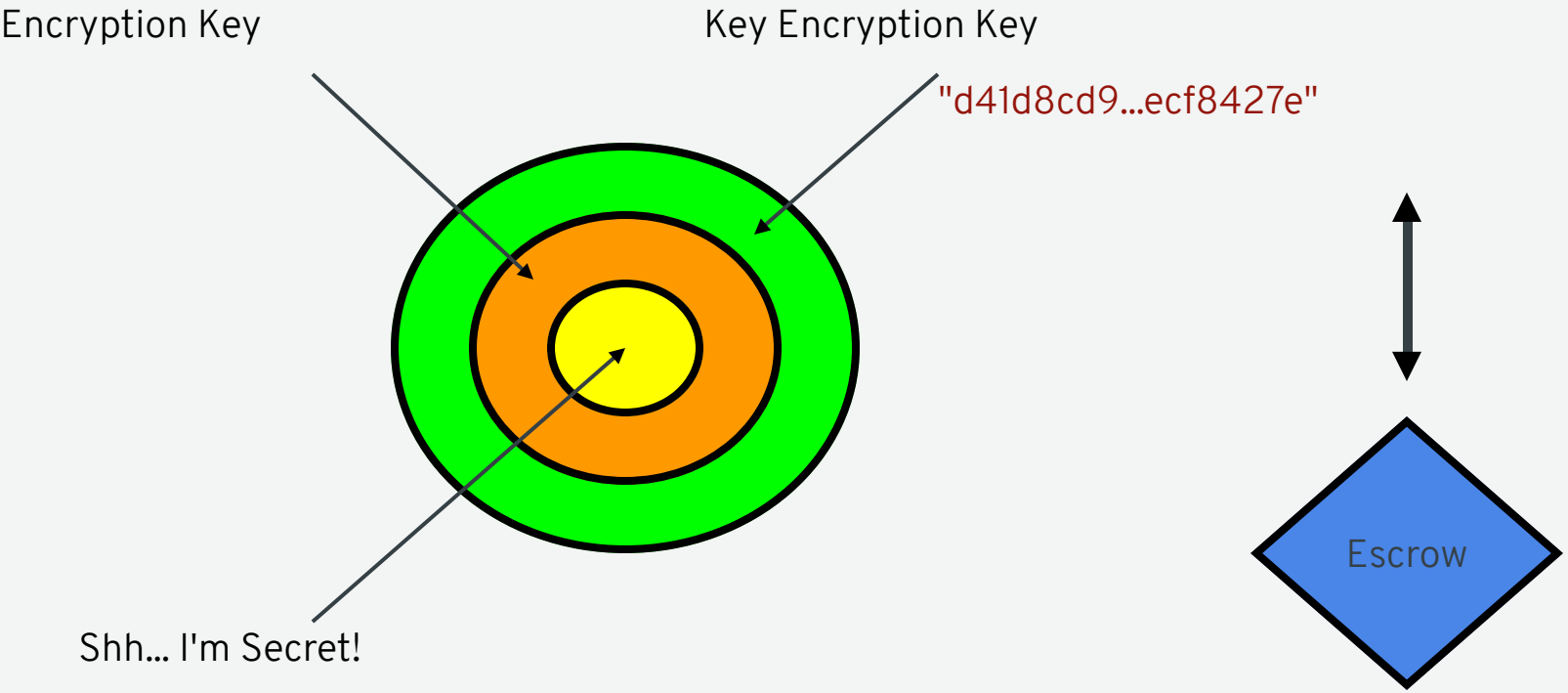

Encryption Key
Key Encryption Key
Shh... I'm Secret!


Encryption Key
Key Encryption Key
"correct battery horse staple"
Shh... I'm Secret!

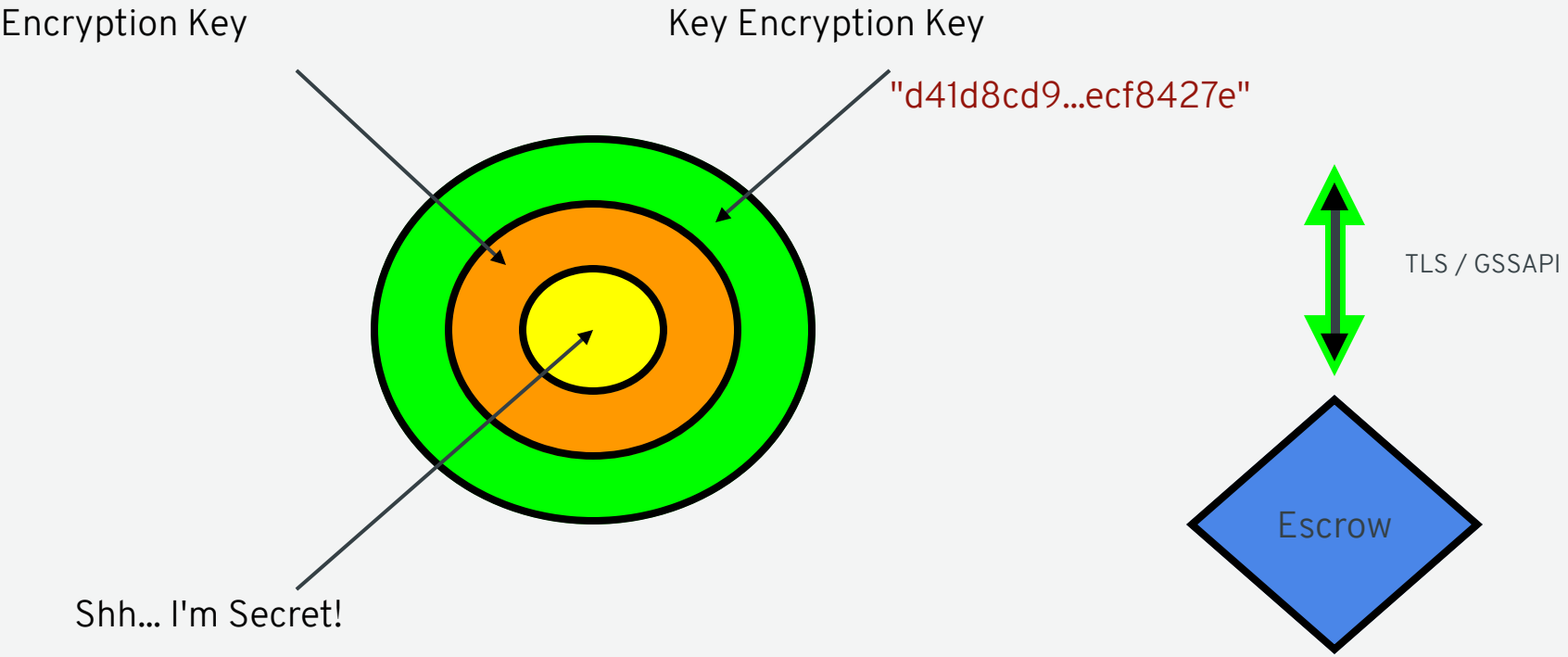
STANDARD PASSWORD MODEL



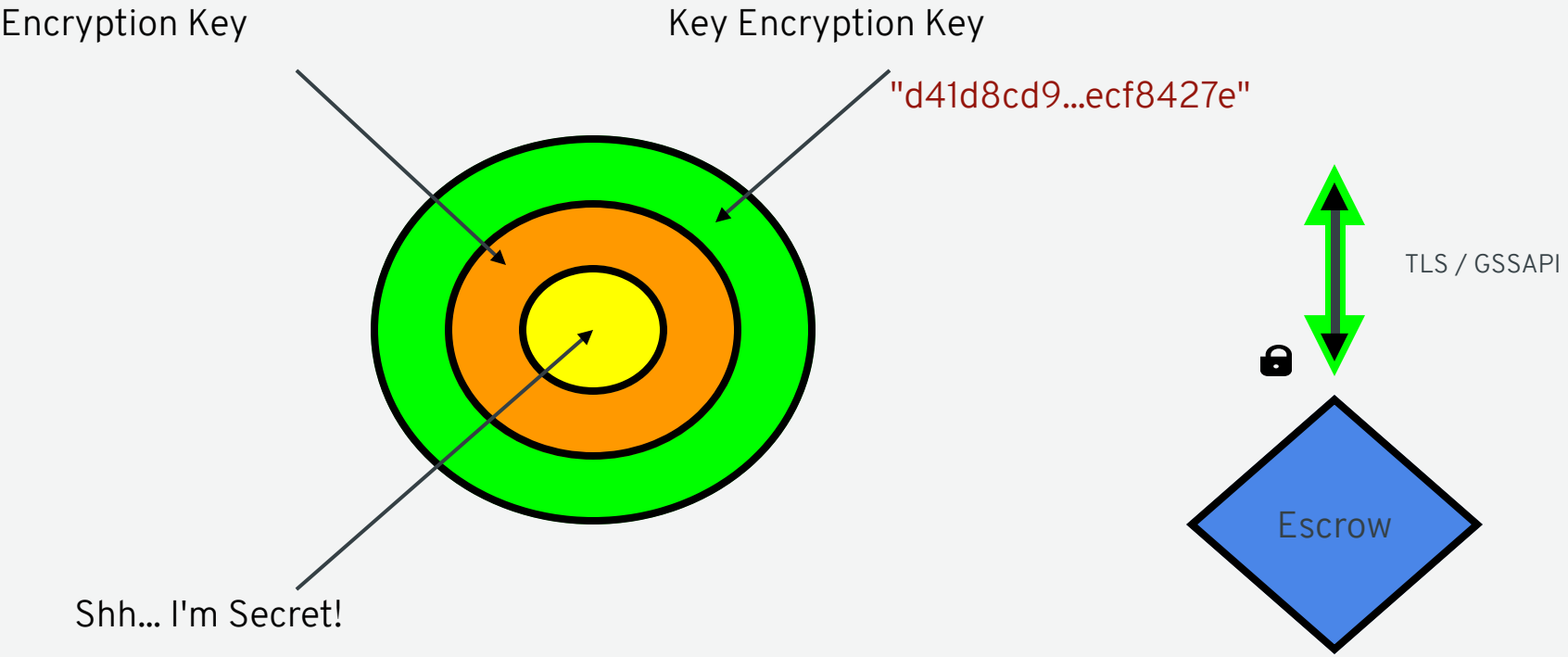
STANDARD ESCROW MODEL?



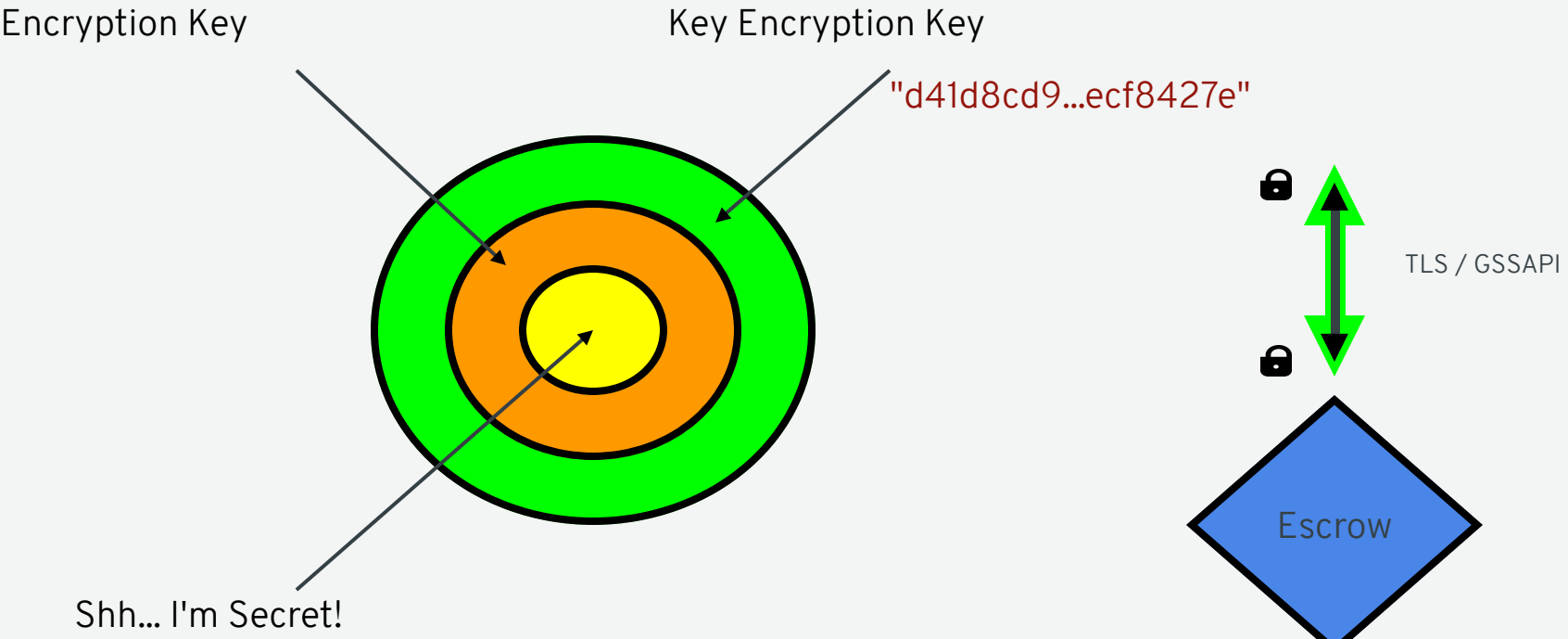
STANDARD ESCROW MODEL?



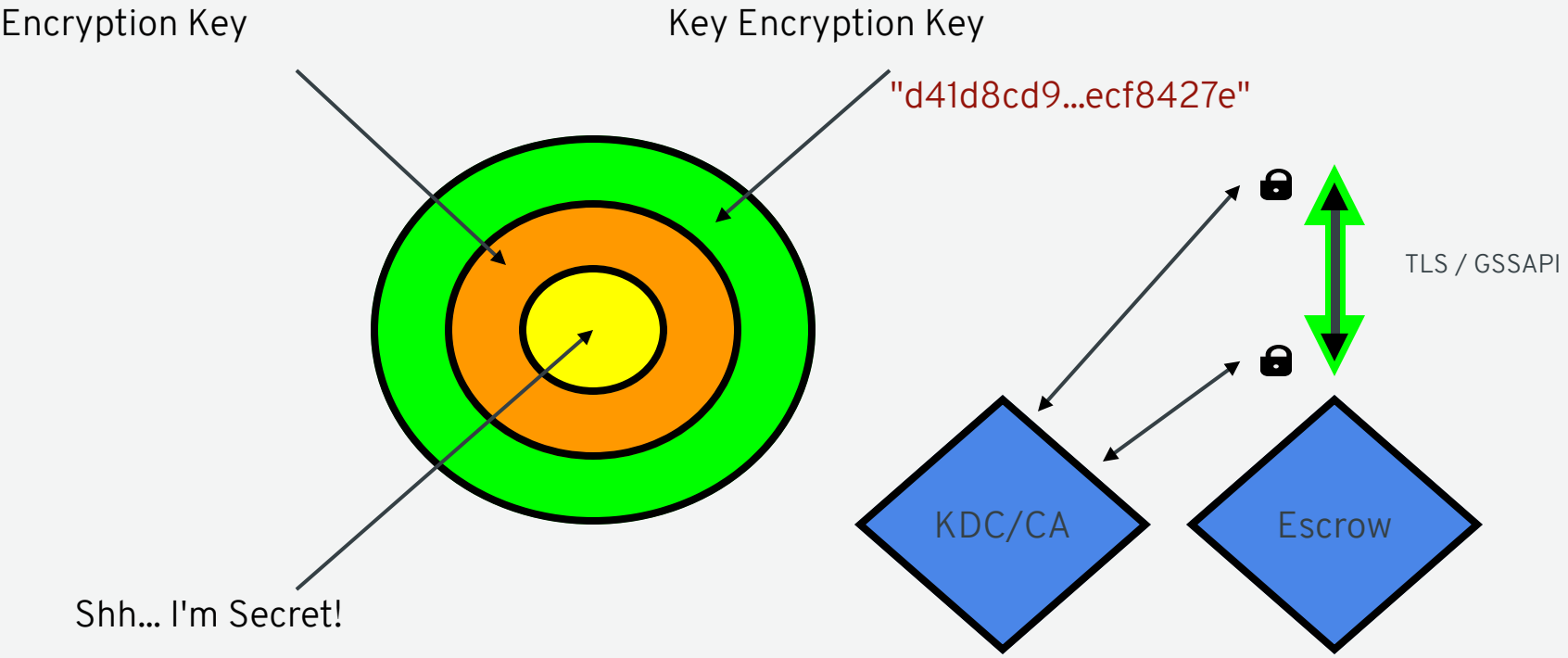
STANDARD ESCROW MODEL?



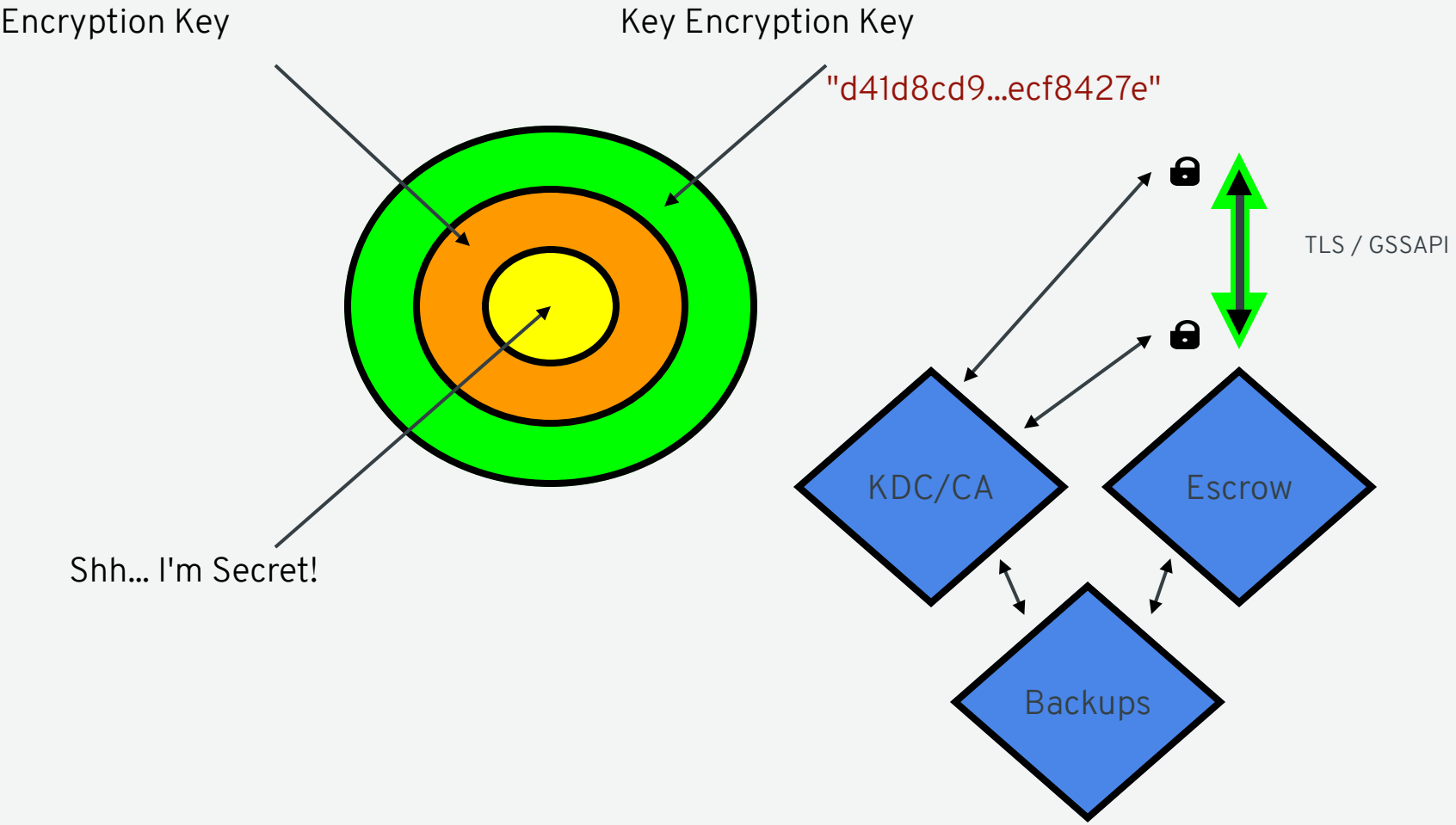
STANDARD ESCROW MODEL?



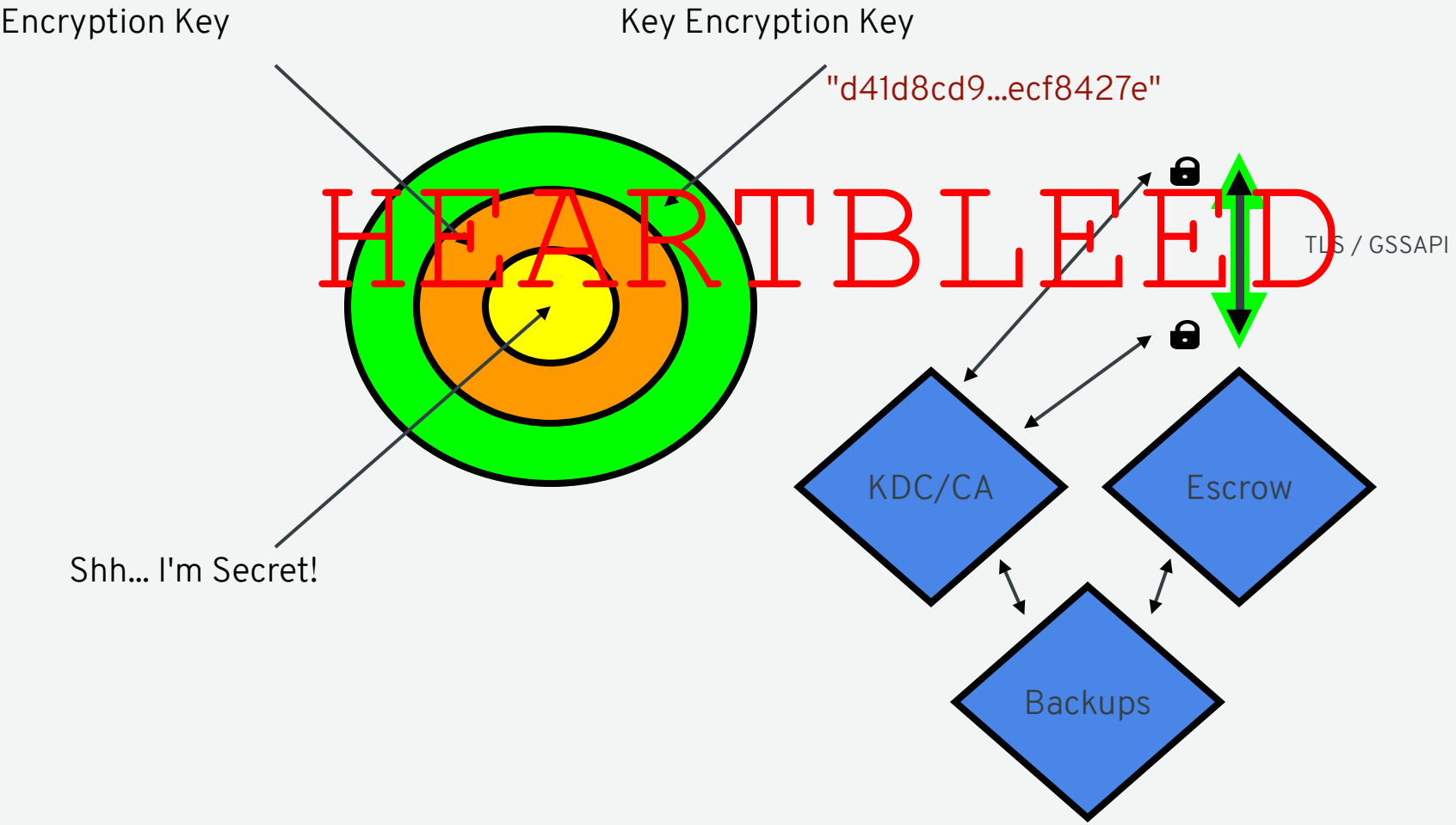
STANDARD ESCROW MODEL?



STANDARD ESCROW MODEL



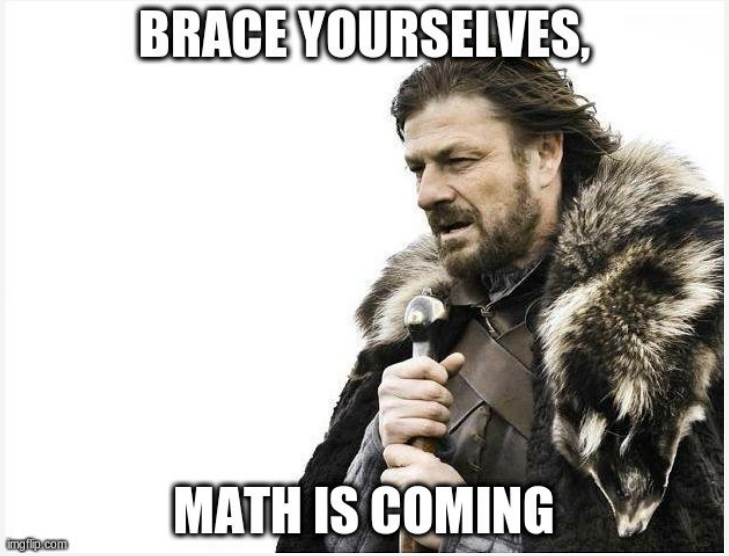
STANDARD ESCROW MODEL



LESSONS LEARNED

- Presuming TLS will protect key transfer is dangerous
- Complexity increases attack surface
- Escrows are difficult to deploy
- X.509 is hard to get right

ASYMMETRIC CRYPTO?


BRACE YOURSELVES,
MATH IS COMING
imgflip.com

(EC) DIFFIE-HELLMAN KEY EXCHANGE

$$
\begin{array}{l|r}
C \in_R [1, p-1] & S \in_R [1, p-1] \\
c = gC & s = gS \\
\qquad\qquad c \longrightarrow & \longleftarrow s \\
K = gSC = sC & K = gCS = cS
\end{array}
$$

BINDING WITH ECDH (INSECURE)

PROVISIONING

$S \in_R [1, p-1]$

$s = gS$

$\longleftarrow s$

$C \in_R [1, p-1]$

$c = gC$

$K = gSC = sC$

$Discard : K, C$

$Retain : s, c$

RECOVERY

$c \longrightarrow$

$K = xS$

$\longleftarrow K$

Weaknesses:

1. K is revealed to a passive attacker.
2. With c, the passive attacker can get K.
3. Server learns c and therefore K.

Resolved: c **MUST** be private

MCCALLUM-RELYEA KEY EXCHANGE

PROVISIONING

$$S \in_R [1, p-1]$$

$$s = gS$$

$$\longleftarrow s$$

$$C \in_R [1, p-1]$$

$$c = gC$$

$$K = gSC = sC$$

$$Discard : K, C$$

$$Retain : s, c$$

RECOVERY

$$E \in_R [1, p-1]$$

$$e = gE$$

$$x = c + e$$

$$x \longrightarrow$$

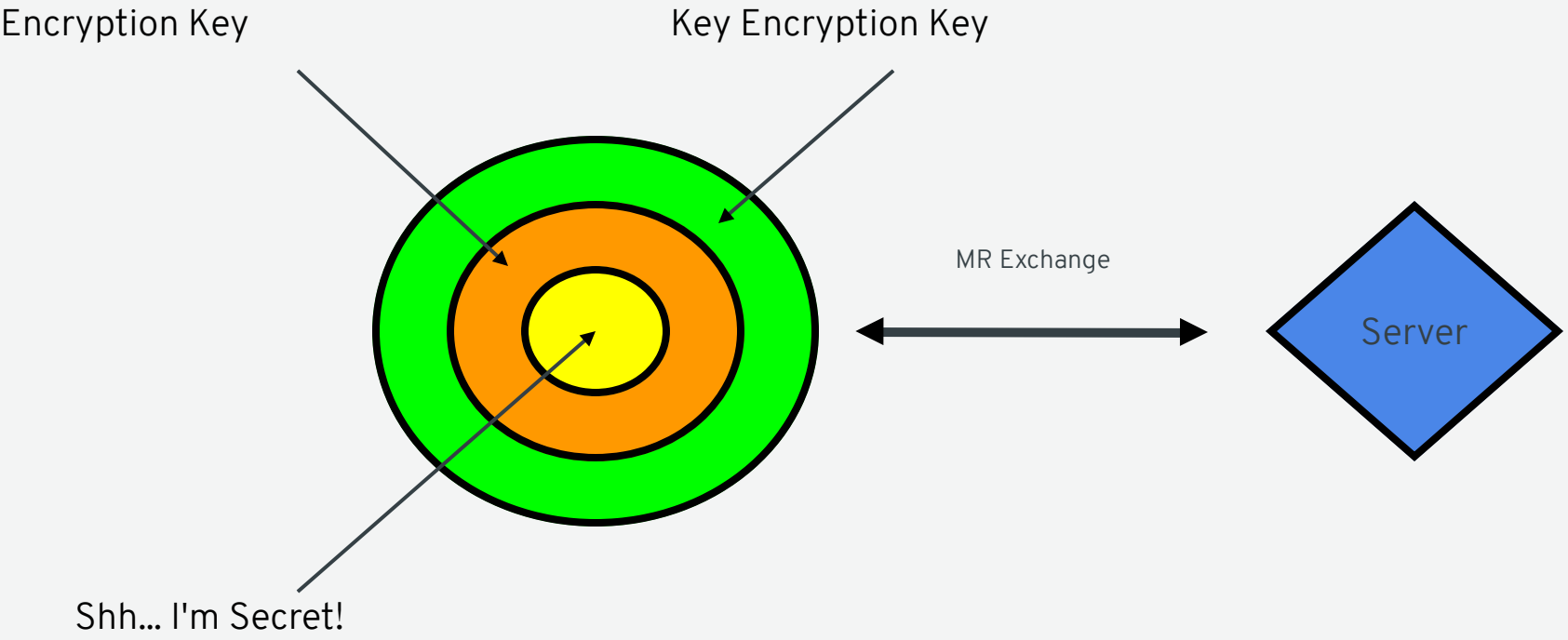
$$y = xS$$

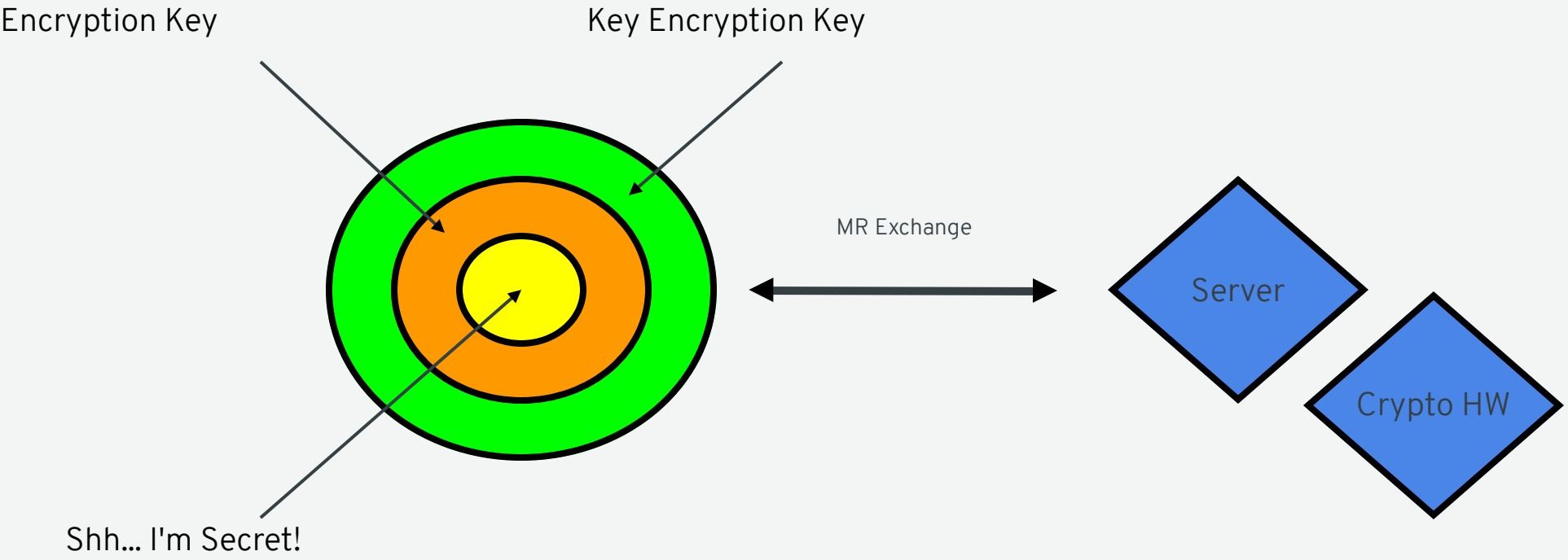
$$\longleftarrow y$$

$$K = y - sE$$

$$Because : K = gCS + gES - gSE$$

To keep c private, e & E **MUST** be private.


Encryption Key
Key Encryption Key
MR Exchange
Server
Shh... I'm Secret!


Encryption Key
Key Encryption Key
MR Exchange
Server
Crypto HW
Shh... I'm Secret!

Property	Escrow	MR Exchange
Server presence during provisioning	Required	Optional
Server presence during recovery	Required	Required
Server knowledge of keys	Required	None
Key transfer	Required	None
Client authentication	Required	Optional
Transport encryption	Required	Optional
End-to-end Encryption	Difficult	Unneeded

TANG

- https://github.com/latchset/tang
- Server-side daemon
- Simple: HTTP + JOSE
- Fast (>2k req/sec)
- Extremely small
- Minimal dependencies
- RHEL 7.4

INSTALLING A TANG SERVER

```
$ sudo yum install tang
$ sudo systemctl enable --now tangd.socket
```

CLEVIS

- https://github.com/latchset/clevis/
- Decryption automation and policy framework
- Minimal dependencies
- Early boot integration
- GNOME integration
- RHEL 7.4

BASIC ENCRYPTION WITH TANG / LUKSV1

```
$ yum install clevis

$ echo redhat | clevis encrypt tang '{"url":"http://localhost"}' > mydata.jwe
The advertisement is signed with the following keys:
        haD7Y-8VkAyJo6-vdZMrGQXCSfI

Do you wish to trust the advertisement? [yN] y

$ cat mydata.jwe
{"ciphertext":"-O59czAqybvxHdme2t3I5A", ...}

$ clevis decrypt < mydata.jwe
redhat

$ systemctl stop tangd.socket

$ clevis decrypt < mydata.jwe
$ echo $?
1
```

DISK BINDING WITH TANG

```
$ clevis bind luks -d /dev/sda1 tang '{"url":"http://tang.srv"}'
The advertisement is signed with the following keys:
        haD7Y-8VkAyJo6-vdZMrGQXCSfI

Do you wish to trust the advertisement? [yN] y
Enter passphrase for /dev/sda1:

$ luksmeta show -d /dev/sda1
0   active empty
1   active cb6e8904-81ff-40da-a84a-07ab9ab5715e
2 inactive empty
3 inactive empty
...

# For root volume unlocking at boot:
$ yum install clevis-dracut
$ dracut -f
$ reboot

# For removable storage GNOME unlocking:
$ yum install clevis-udisks2
```

KNOWN ISSUES

dracut / NetworkManager integration problems

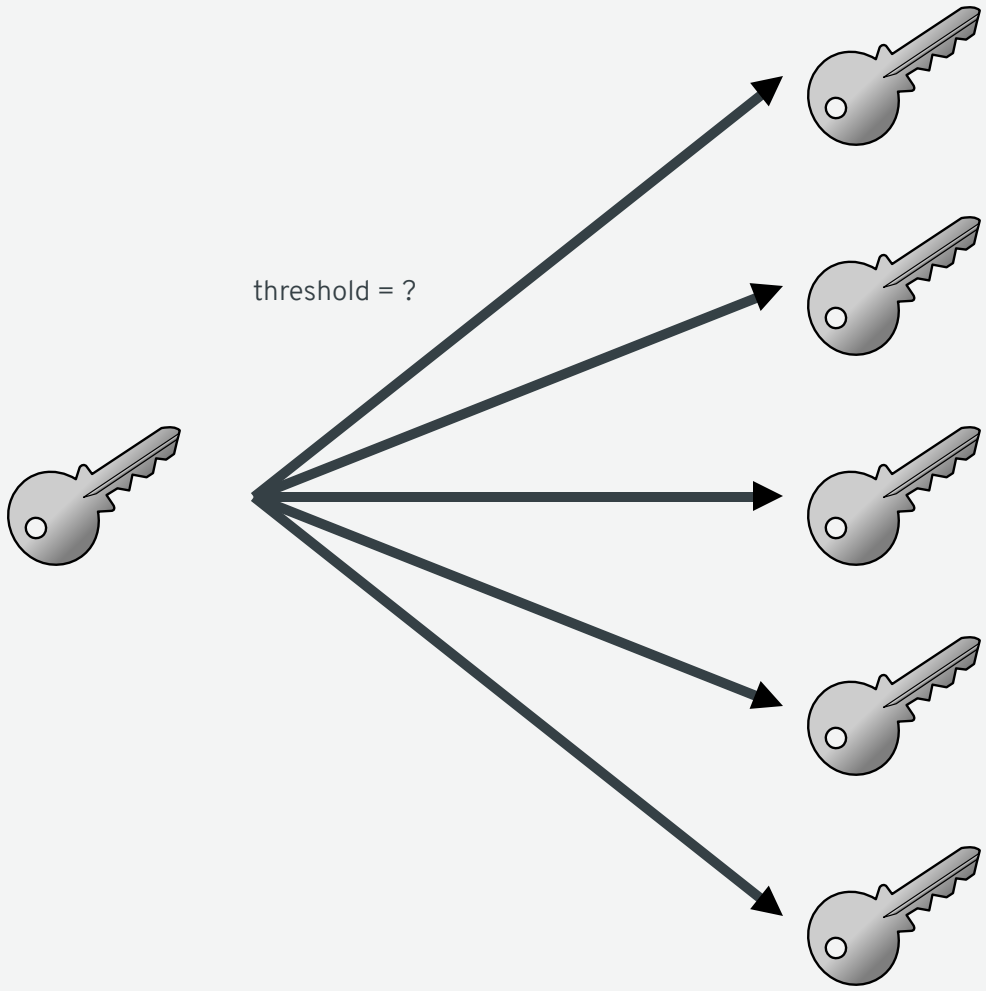
- Active BZ with fix pending
- Shouldn't hurt in most cases, but it looks bad

```
[root@nbde7 ~]$ ip route
default via 192.168.122.1 dev eth0
default via 192.168.122.1 dev eth0 proto dhcp metric 100
192.168.122.0/24 dev eth0 proto kernel scope link src 192.168.122.170
192.168.122.0/24 dev eth0 proto kernel scope link src 192.168.122.170 metric 100
```

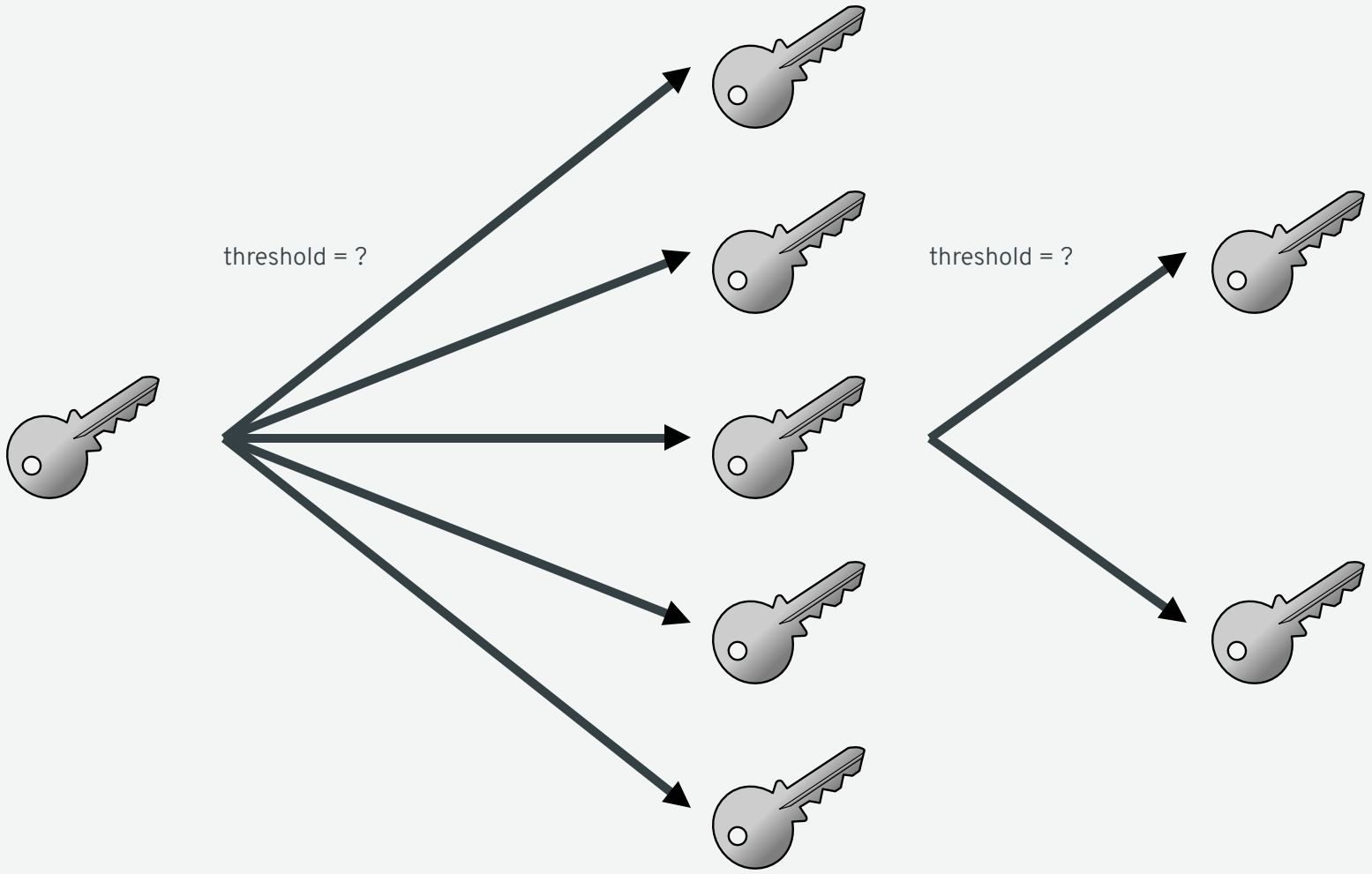
FROM AUTOMATION TO POLICY

YESTERDAY	TODAY	TOMORROW
Standards (AES, PCI-DSS, etc.)	Automation	Policy

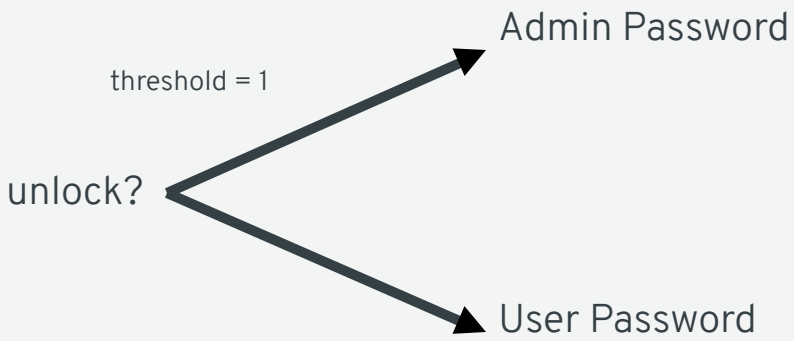
SHAMIR SECRET SHARING



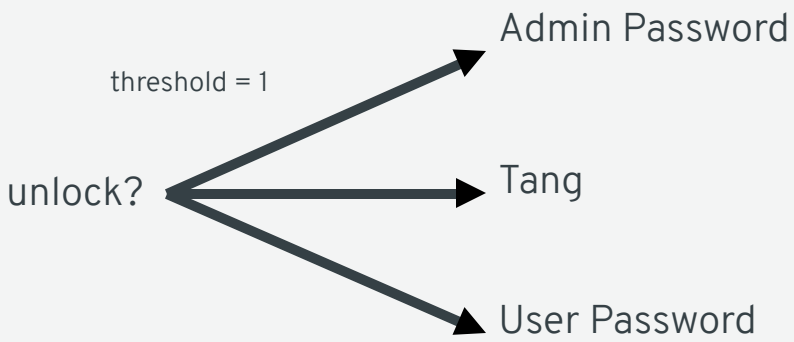
SHAMIR SECRET SHARING



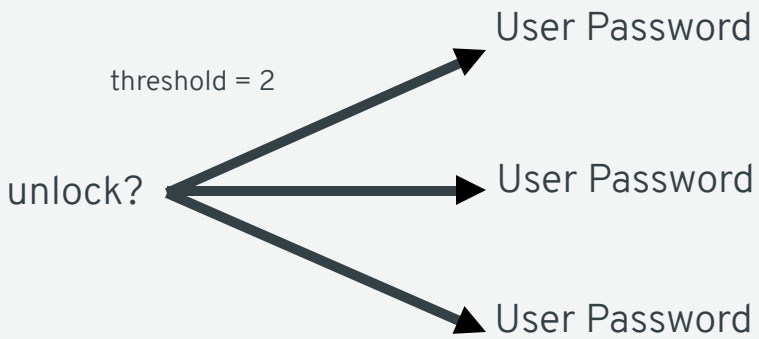
SIMPLE LAPTOP



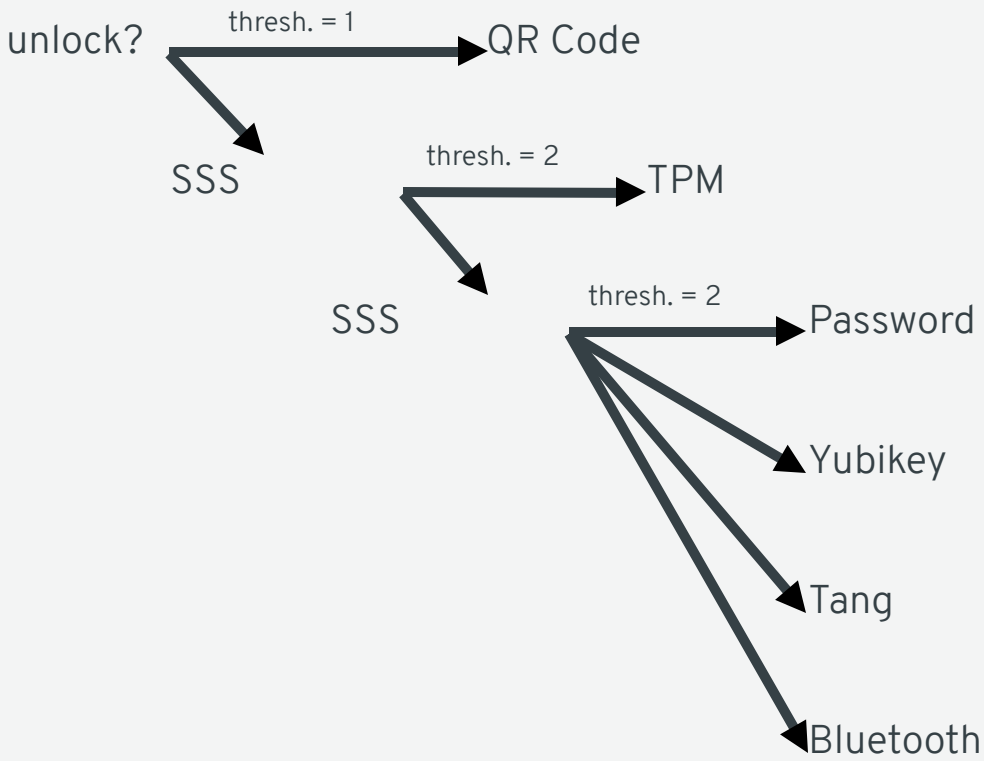
AUTOMATED LAPTOP



HIGH SECURITY SYSTEM



COMPLEX LAPTOP POLICY



BASIC SHAMIR'S WITH TANG

```
$ echo PT | clevis encrypt sss \
'{"pins": {"tang": [{"url": "http://a.tang.srv"}, {"url": "http://b.tang.srv"}]}, "t": 1}' \
> out.jwe
The advertisement is signed with the following keys:
        haD7Y-8VkAyJo6-vdZMrGQXCSfI

Do you wish to trust the advertisement? [yN] y

The advertisement is signed with the following keys:
        Edp-ESShUx4_95kGt-DTsCBbPag

Do you wish to trust the advertisement? [yN] y

$ clevis decrypt < out.jwe
PT

# Bring Down Tang Server A
$ clevis decrypt < out.jwe
PT

# Bring Down Tang Server B
$ clevis decrypt < out.jwe
$ echo $?
1
```

EXPLORING THE ECOSYSTEM

DEPENDENCY: JOSÉ

- https://github.com/latchset/jose
- JSON Object Signing and Encryption
- C Library & Command Line Utility
- Bottom Line: User-Friendly, Standards Compliant Crypto

```
$ jose jwk gen -i '{"alg": "A128GCM"}' -o oct.jwk
$ jose jwk gen -i '{"alg": "RSA1_5"}' -o rsa.jwk
$ jose jwk gen -i '{"alg": "ES256"}' -o ec.jwk

$ echo hi | jose jwe enc -i- -k rsa.pub.jwk -o msg.jwe
$ jose jwe dec -i msg.jwe -k rsa.jwk
hi
$ jose jwe dec -i msg.jwe -k oct.jwk
Decryption failed!

$ echo hi | jose jws sig -i- -k ec.jwk -o msg.jws
$ jose jws ver -i msg.jws -k ec.pub.jwk
hi
$ jose jws ver -i msg.jws -k oct.jwk
No signatures validated!
```

DEPENDENCY: LUKSMETA

- https://github.com/latchset/luksmeta
- Store metadata in LUKSv1 header gap
- C library & Command Line Utility

```
$ echo hi | luksmeta save -d /dev/sdc1 -s 2 -u EC998562-B60D-47F0-A579-DCA8C12F5BF6

$ luksmeta load -d /dev/sdc1 -s 2 -u EC998562-B60D-47F0-A579-DCA8C12F5BF6
hi

$ luksmeta load -d /dev/sdc1 -s 2 -u 12618962-A1E5-48F1-B327-D7C60E20FC02
Slot contains different UUID
```

FUTURE FEATURES

JOSÉ

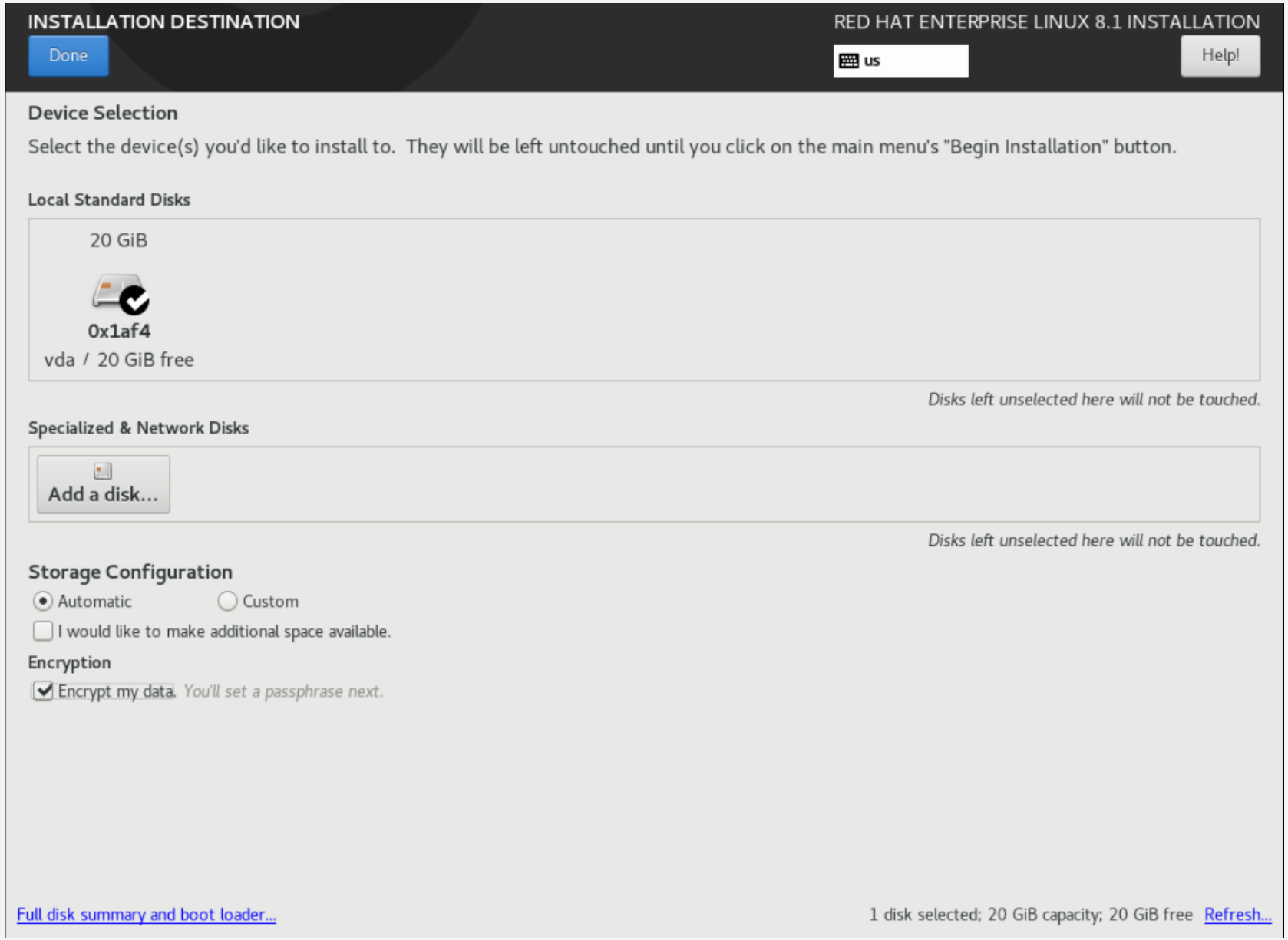
- PKCS#11 Support
- Python Bindings
- Additional crypto backends
- Additional algorithms

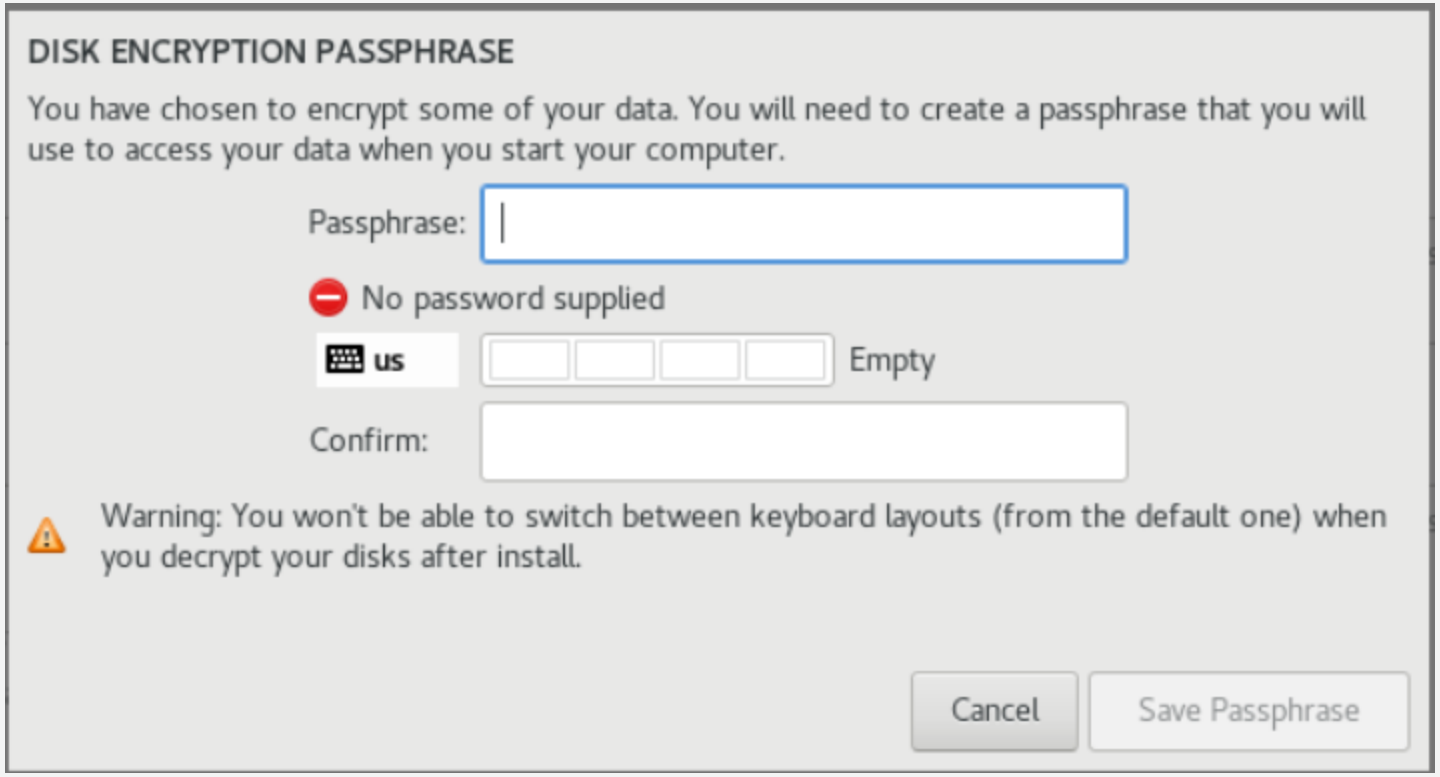
CLEVIS

- Password Pin
- PKCS#11 Pin (including, in the future, TPM)
- Ext4 encryption support

TANG

- Binding IDs (Optional; sacrifices anonymity)
- Revocation (requires Binding IDs)


INSTALLATION DESTINATION
Done
RED HAT ENTERPRISE LINUX 8.1 INSTALLATION
us
Help!
Device Selection
Select the device(s) you'd like to install to. They will be left untouched until you click on the main menu's "Begin Installation" button.
Local Standard Disks
20 GiB
0x1af4
vda / 20 GiB free
Disks left unselected here will not be touched.
Specialized & Network Disks
Add a disk...
Disks left unselected here will not be touched.
Storage Configuration
Automatic
Custom
I would like to make additional space available.
Encryption
Encrypt my data. You'll set a passphrase next.
Full disk summary and boot loader...
1 disk selected; 20 GiB capacity; 20 GiB free
Refresh...


DISK ENCRYPTION PASSPHRASE
You have chosen to encrypt some of your data. You will need to create a passphrase that you will use to access your data when you start your computer.
Passphrase:
No password supplied
us
Empty
Confirm:
Warning: You won't be able to switch between keyboard layouts (from the default one) when you decrypt your disks after install.
Cancel
Save Passphrase

DEMO

RESOURCES

- RHEL 7:
  - https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Security_Guide/sec-Using_Network-Bound_Disk_Encryption.html
- RHEL 8:
  - https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/security_hardening/configuring-automated-unlocking-of-encrypted-volumes-using-policy-based-decryption_security-hardening
- Multi-device setup
  - https://access.redhat.com/articles/4500491

QUESTIONS?