Adam Zygmontowicz, Micah Thornton, & Jennifer Dworak

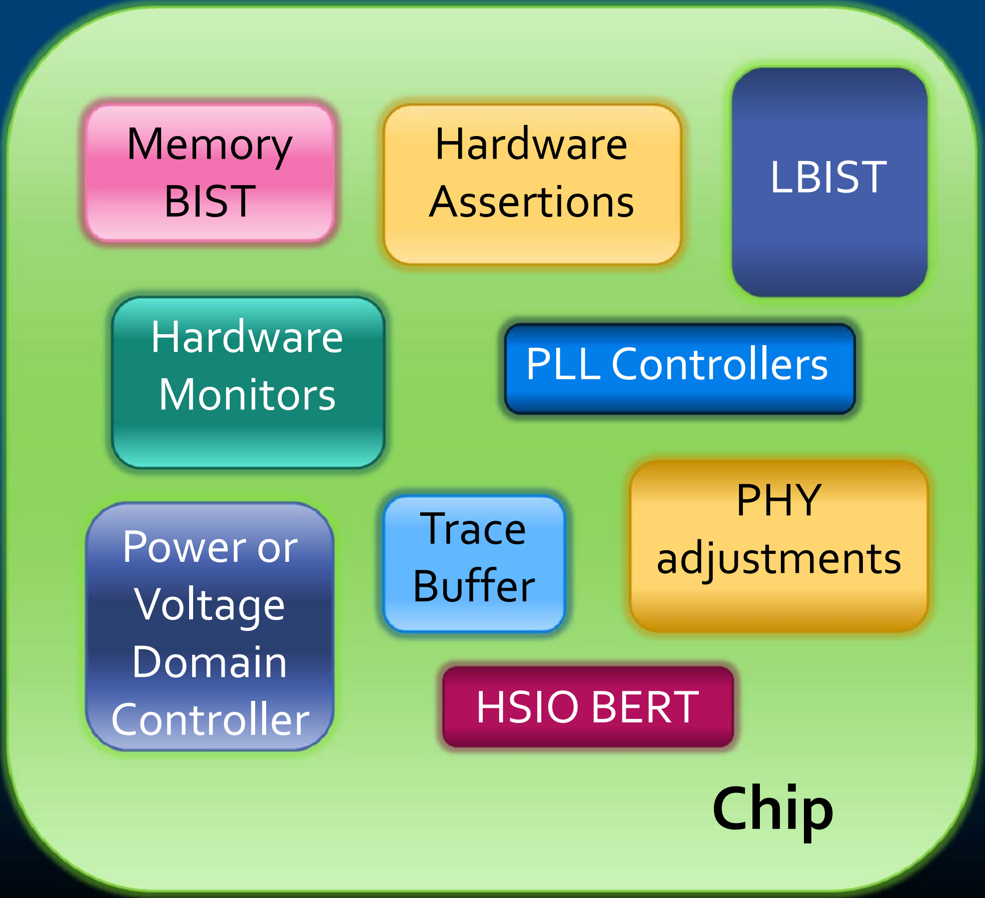
Al Crouch & John Potter

SECURING YOUR SYSTEM WITH P1687

Presented to the 2013 Nordic Test Forum

Motivation: On-Chip Instruments Need Access Protection

- Chips may contain multiple cores and hundreds of instruments
- Depending on the design, many of these can often be accessed through scan chains



On-Chip Data Also Needs Protection from Unauthorized Access

- Chip IDs
- Encryption Keys
- DVD codes
- Intermediate Execution States
- On-chip IP accessible through scan chain

We need a way to prevent unauthorized users from accessing this information through the scan chain.

So, how *can* we protect things that must not be accessed?

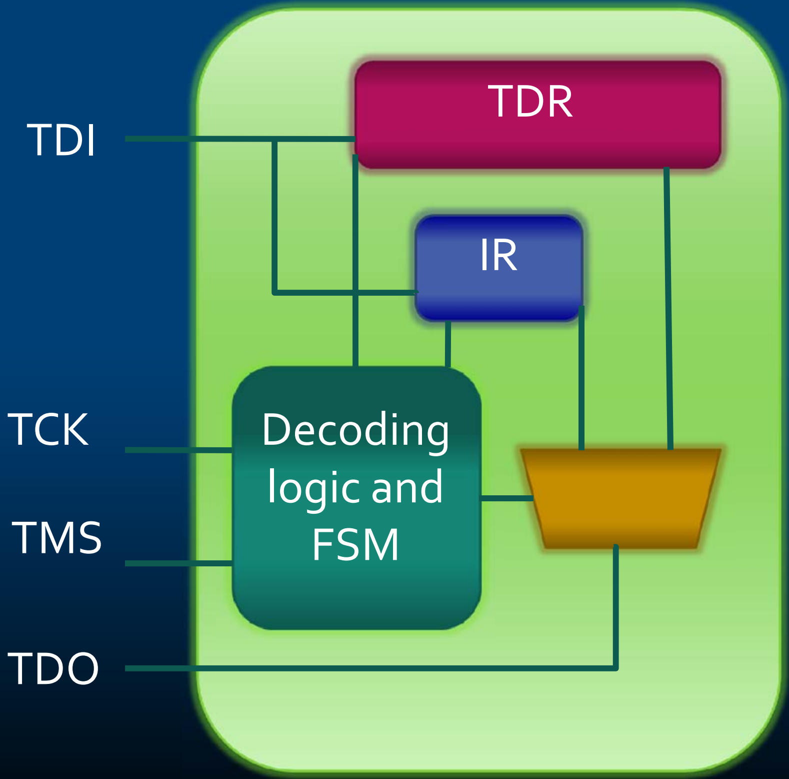
It depends on how you normally intend to access them in the first place.

Outline

- Protecting Standard JTAG
- Overview of P1687
- Hiding Instruments behind LSIBs
- Adding Traps for Extra Protection
- Distributing Locks, Keys, and Traps across Multiple Levels of Hierarchy
- Conclusions & Future Work

Internals of chip/board can be accessed with IEEE 1149.1

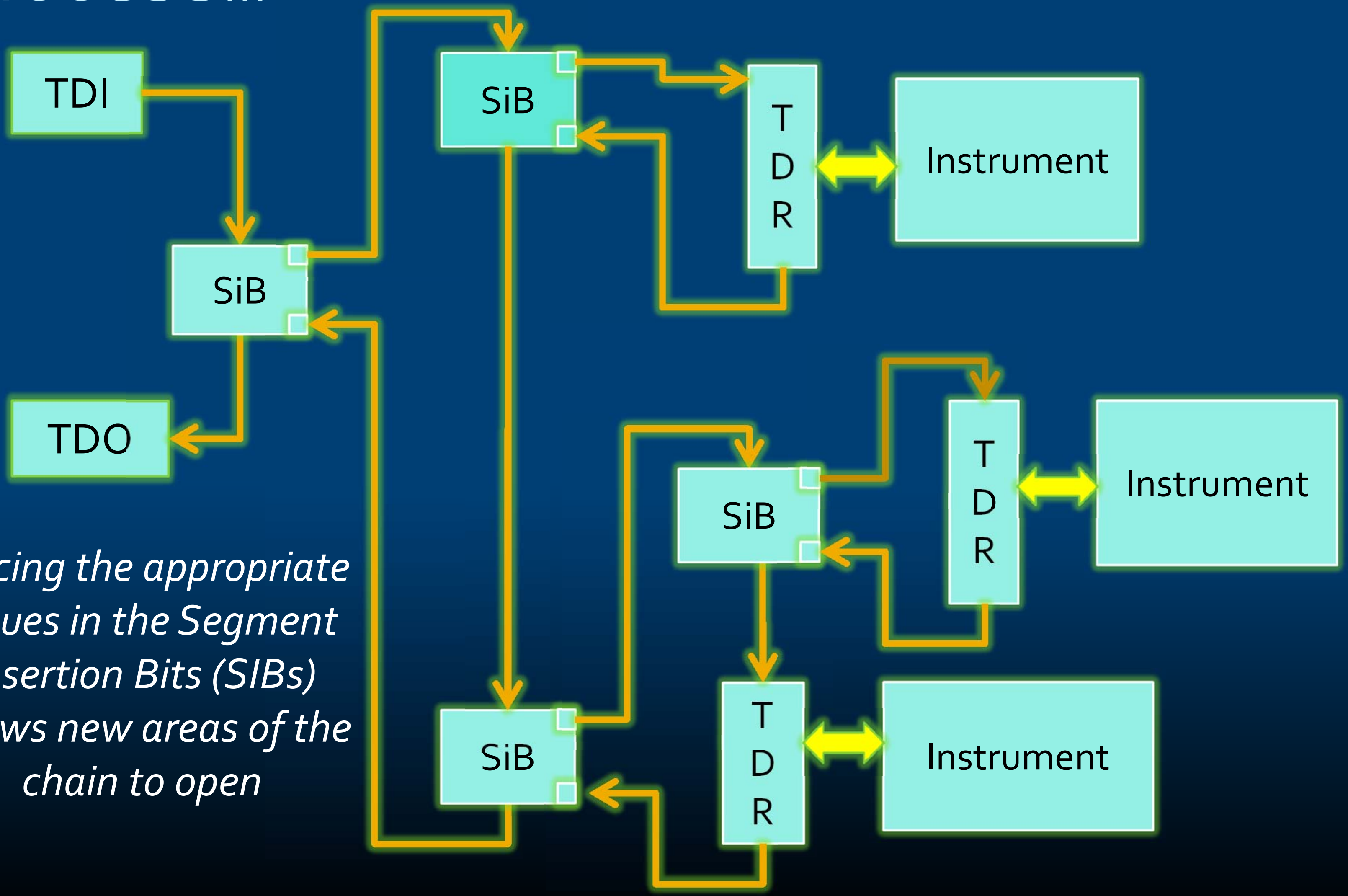
- Attackers can hack the JTAG port
  - Modify firmware (Xbox Hack)
  - Search for "private" instructions that are hidden test and debug features
  - Altera lists private instructions to avoid or you will destroy the chip!



How can information be protected in 1149.1?

- Burn out the JTAG port by fusing off the TMS signal.
    - Problem: Can no longer use JTAG for debug or field test.
- Others suggest adding extra hardware to handle challenge/response pairs
    - SHA256 HASH engine used to compute a hash with a secret key. (Clark 2010)
        - Random challenge generated with multiple ring oscillators
        - User must compute same hash
    - Security Authentication Module (SAM) and Access Monitor (AM). (Pierce 2011, 2012)
        - SAM locks JTAG interface on bootup and authentication protocol is used to unlock access and assess privilege level.
        - UpdateDR is blocked if user doesn't have privileges

P1687 is designed for instrument access…

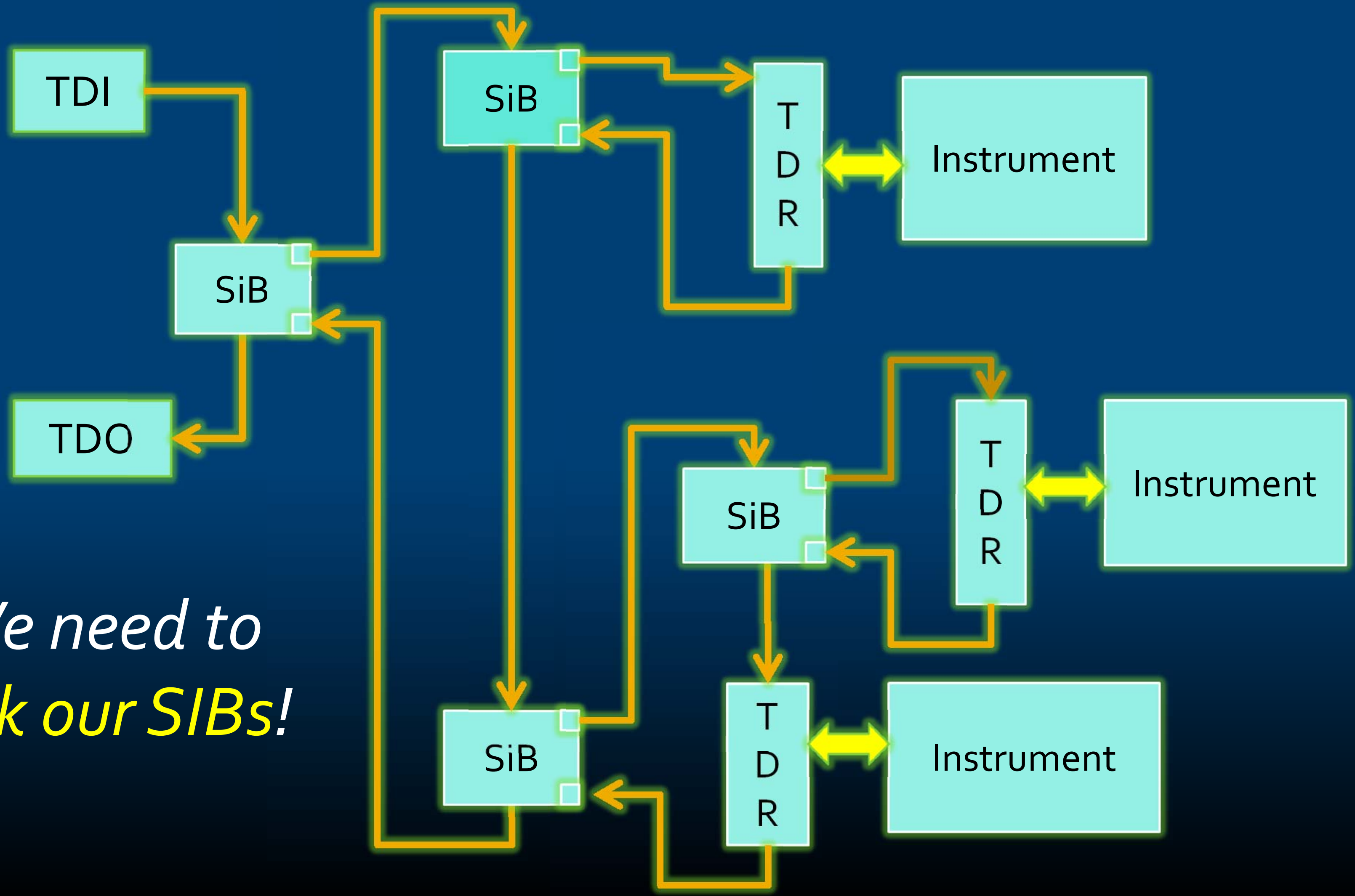


Placing the appropriate values in the Segment Insertion Bits (SIBs) allows new areas of the chain to open

Can we harness the design of a P1687 network to inexpensively increase security by hiding instruments?

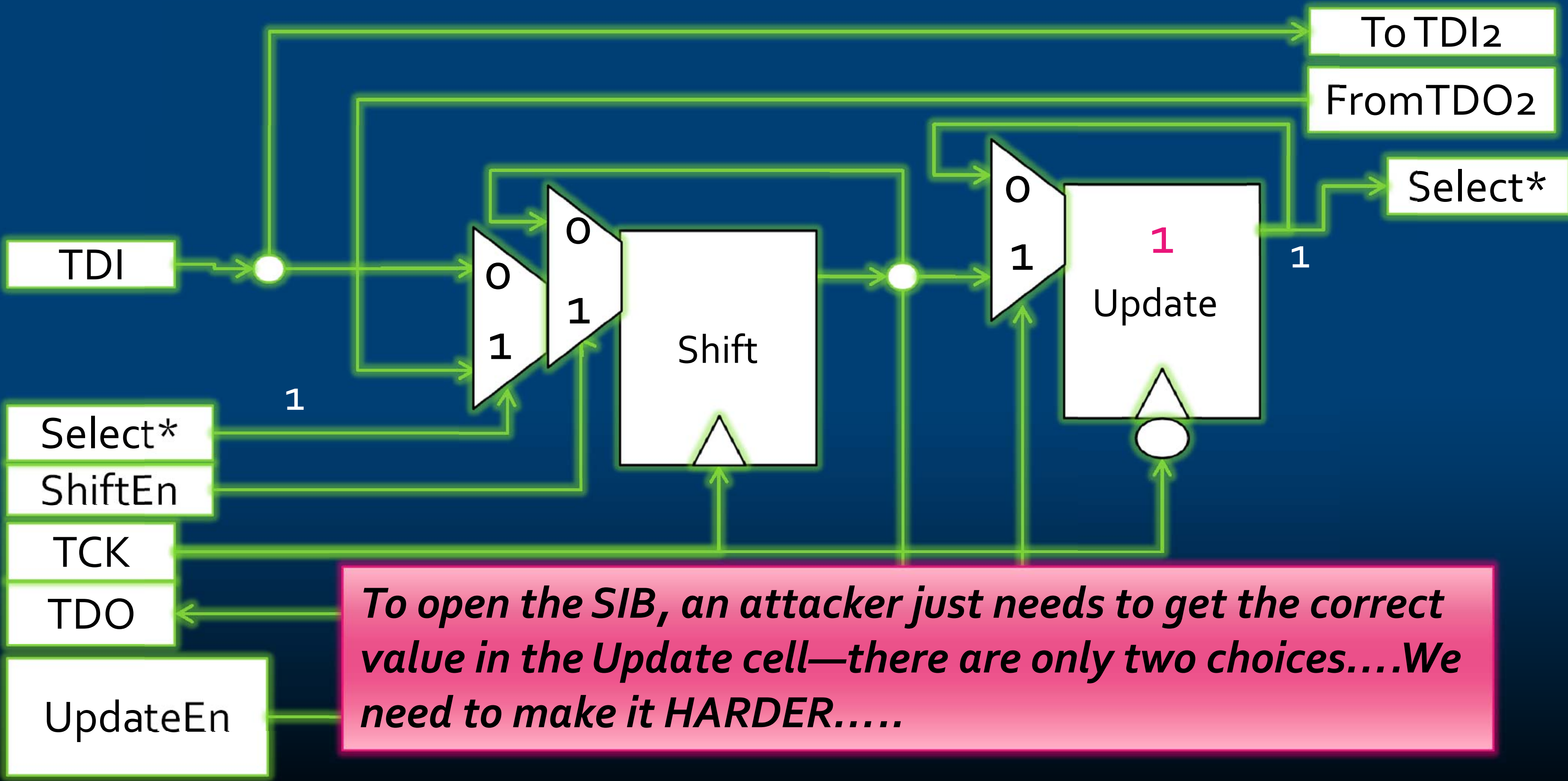
Yes!

SIBs serve as doors that must be opened for instrument access…



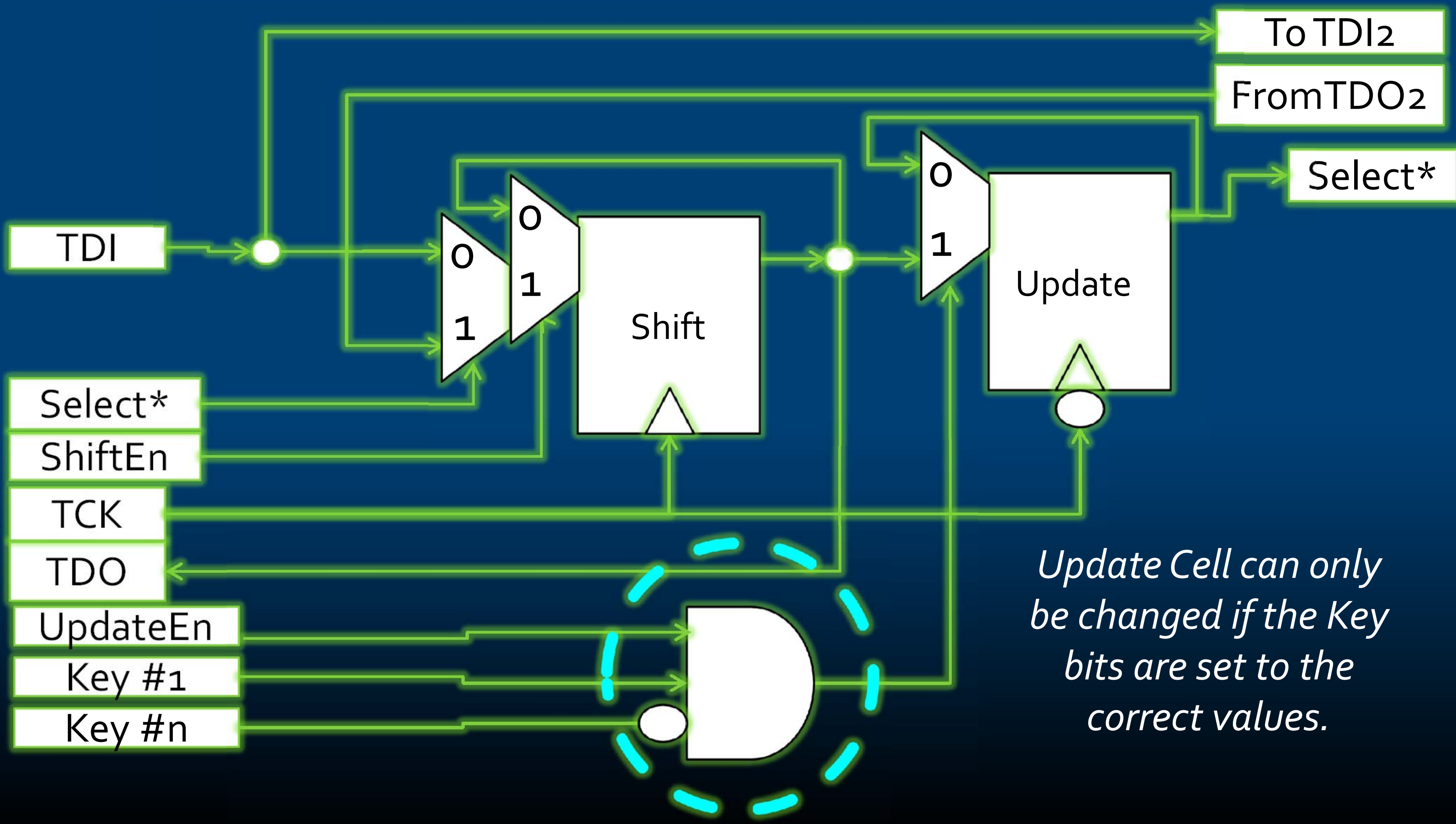
We need to lock our SIBs!

What does a normal SIB look like?



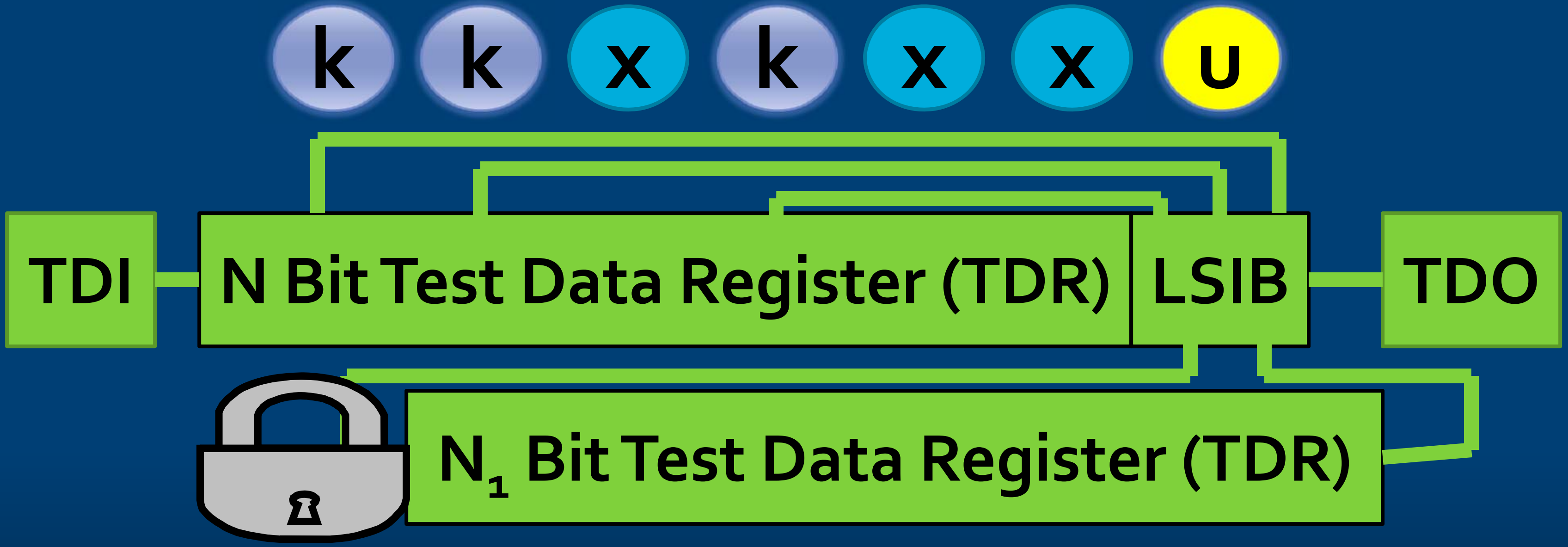
To open the SIB, an attacker just needs to get the correct value in the Update cell—there are only two choices….We need to make it HARDER…..

One Implementation of a Locking Segment Insertion Bit (LSIB)



Update Cell can only be changed if the Key bits are set to the correct values.

The attacker must guess the right key bit values to open the LSIB…


k
k
x
k
x
x
u
TDI
N Bit Test Data Register (TDR)
LSIB
TDO
N_1 Bit Test Data Register (TDR)

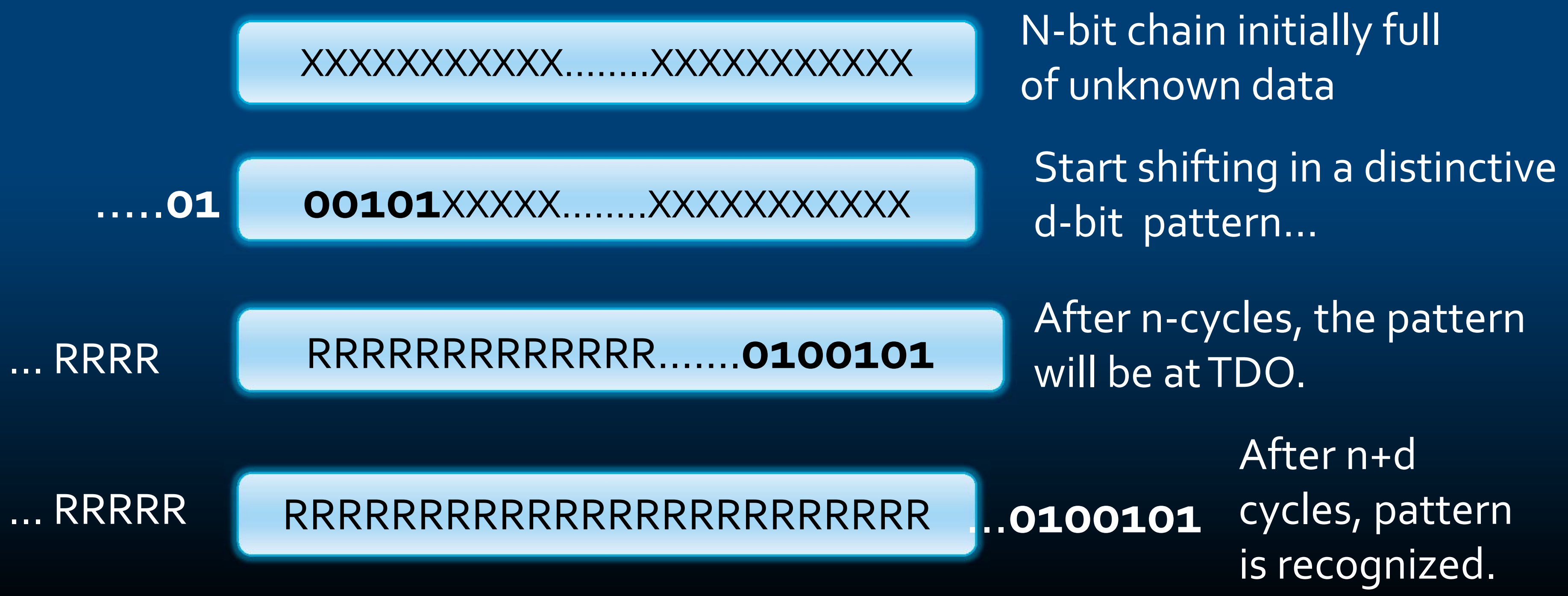
If the attacker doesn't know anything about the network, he can start by trying random guesses….

How long is guessing likely to take?

How will the attacker know that a LSIB has been opened?

The Scan Chain Length generally changes when we open a SIB.....

How will the attacker know the length of the chain?



XXXXXXXXXXX........XXXXXXXXXXX

N-bit chain initially full of unknown data

.....**01** **00101**XXXXX........XXXXXXXXXXX

Start shifting in a distinctive d-bit pattern...

... RRRR RRRRRRRRRRRRR.......**0100101**

After n-cycles, the pattern will be at TDO.

... RRRRR RRRRRRRRRRRRRRRRRRRRRRR ...**0100101**

After n+d cycles, pattern is recognized.

How does the attacker make a guess?

The attacker uses the 1149.1 state machine to try to open the SIB

Update DR	Once a random vector is in the chain, Update DR needed to try to open LSIB: 1 cycle.
Run Test Idle	1 cycle.
Select DR	1 cycle.
Capture DR	1 cycle.
Shift DR	1 cycle.
Shift Cycles	N+d cycles to check the length of the chain and shift in random bits.

Total time for a guess…

5+n+d

How Attackers Unlock LSIBs

Needed Key = 101

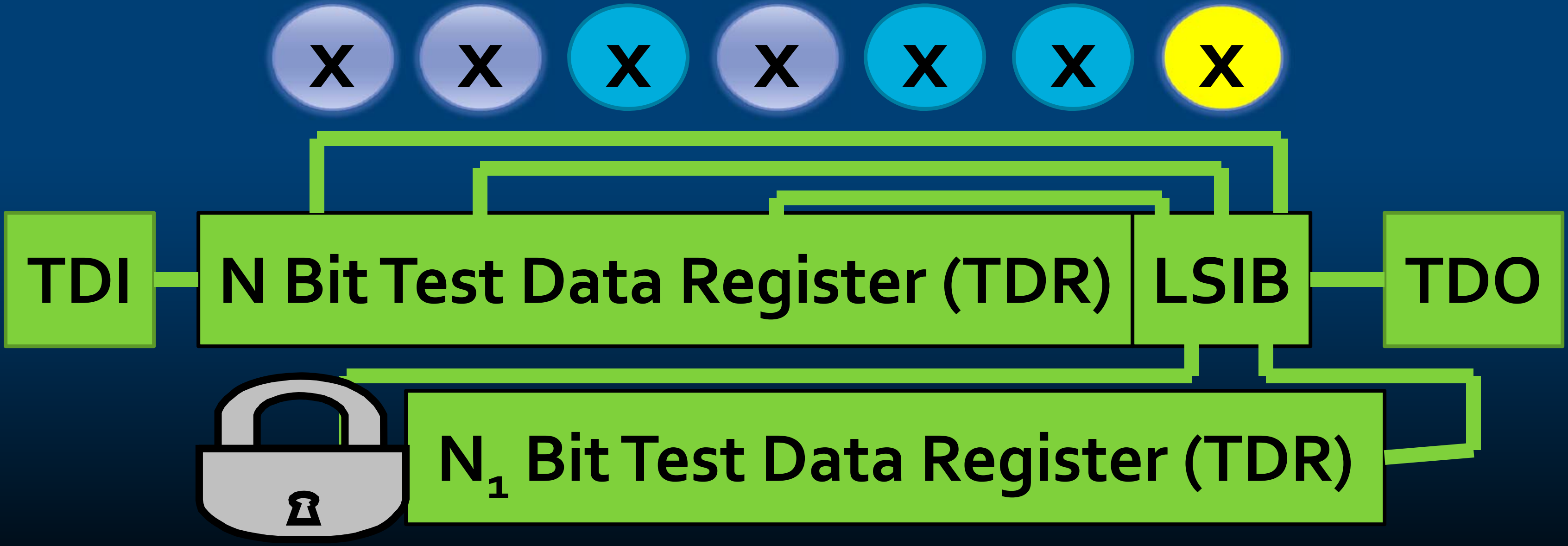
First guess = 1000010

Distinguishing seq = 0100101

● LSIB

● Key

● Data



X
X
X
X
X
X
X
TDI
N Bit Test Data Register (TDR)
LSIB
TDO
N₁ Bit Test Data Register (TDR)

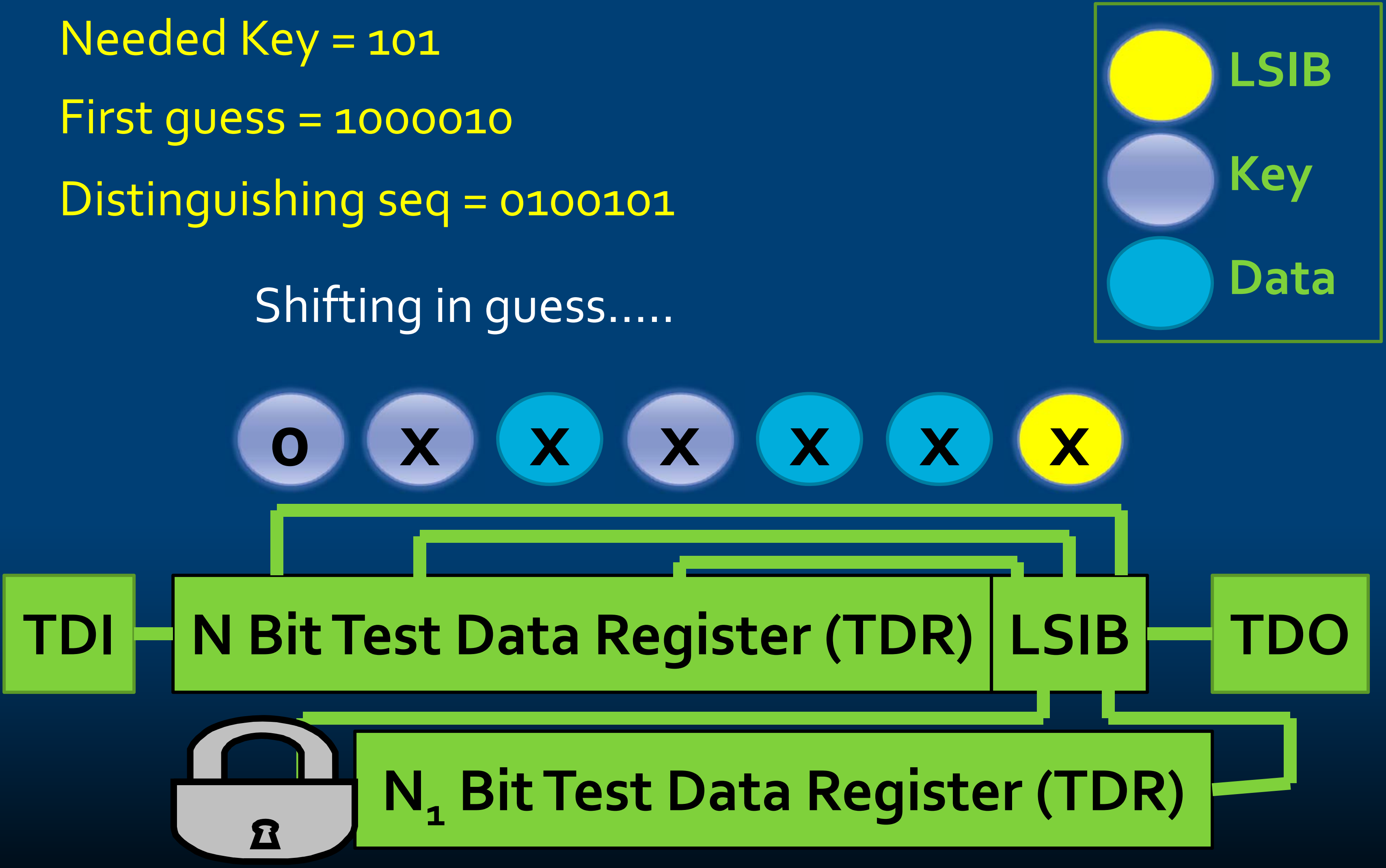
How Attackers Unlock LSIBs

Needed Key = 101

First guess = 1000010

Distinguishing seq = 0100101

Shifting in guess…..


LSIB
Key
Data
0
X
X
X
X
X
X
TDI
N Bit Test Data Register (TDR)
LSIB
TDO
N_1 Bit Test Data Register (TDR)

How Attackers Unlock LSIBs

Needed Key = 101

First guess = 1000010

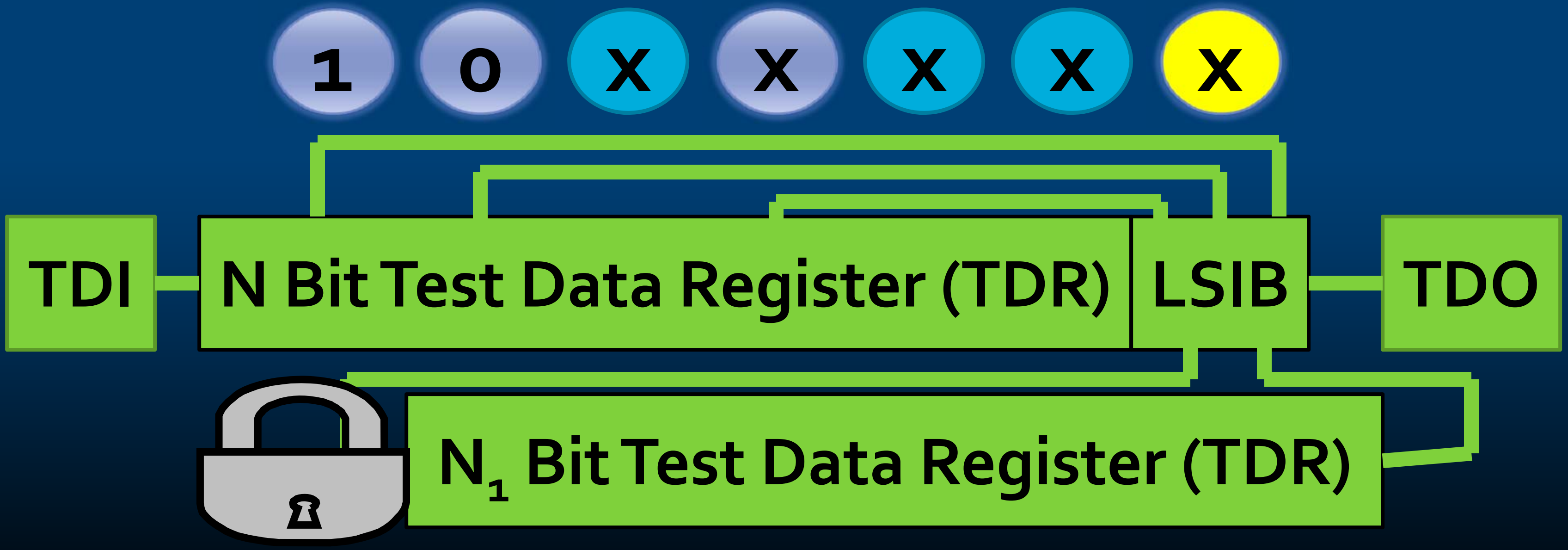
Distinguishing seq = 0100101

Shifting in guess…..

● LSIB

● Key

● Data



1 0 x x x x x

TDI

N Bit Test Data Register (TDR)

LSIB

TDO

N_1 Bit Test Data Register (TDR)

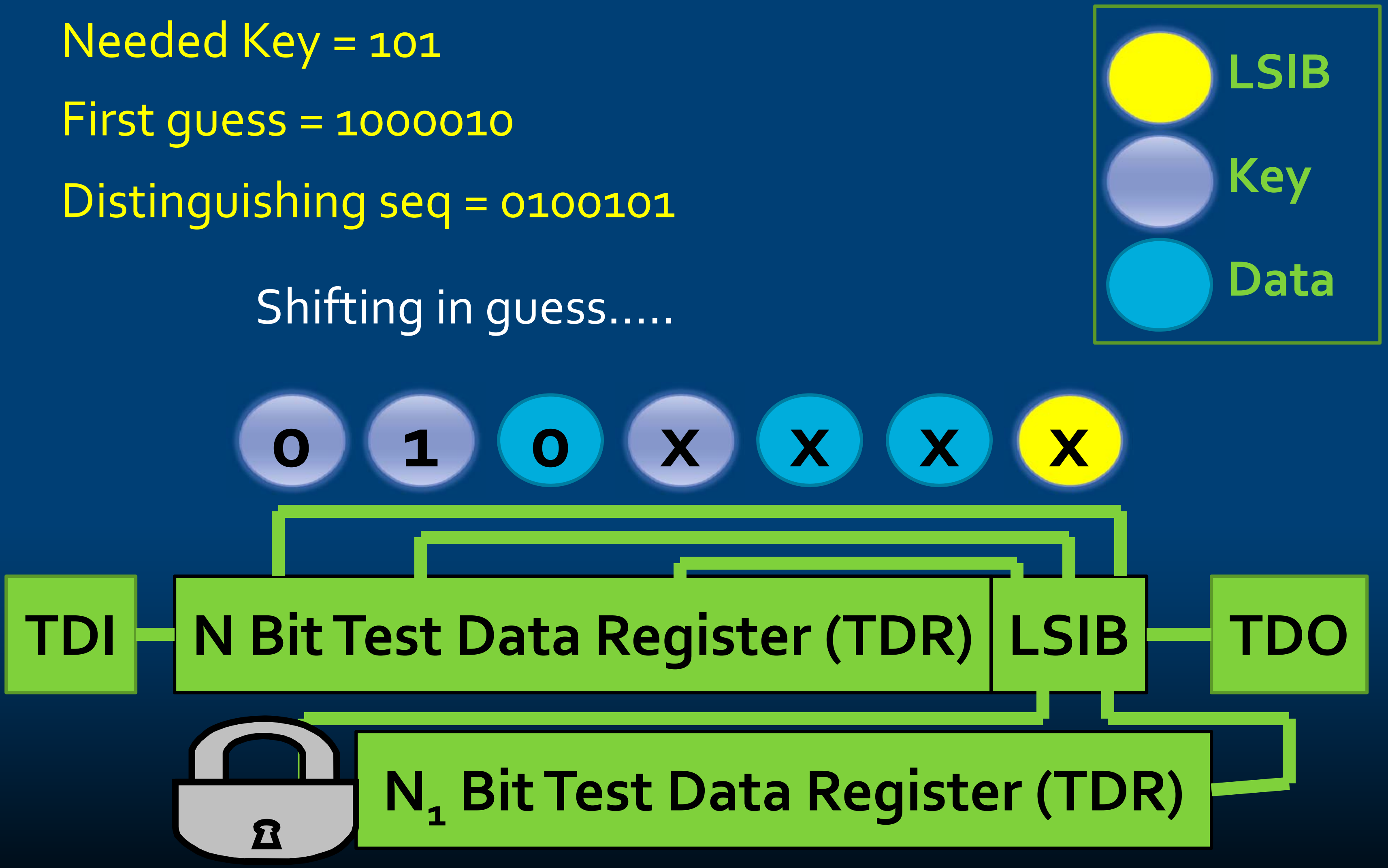
How Attackers Unlock LSIBs

Needed Key = 101

First guess = 1000010

Distinguishing seq = 0100101

Shifting in guess.....



LSIB
Key
Data
0
1
0
X
X
X
X
TDI
N Bit Test Data Register (TDR)
LSIB
TDO
N_1 Bit Test Data Register (TDR)

How Attackers Unlock LSIBs

Needed Key = 101

First guess = 1000010

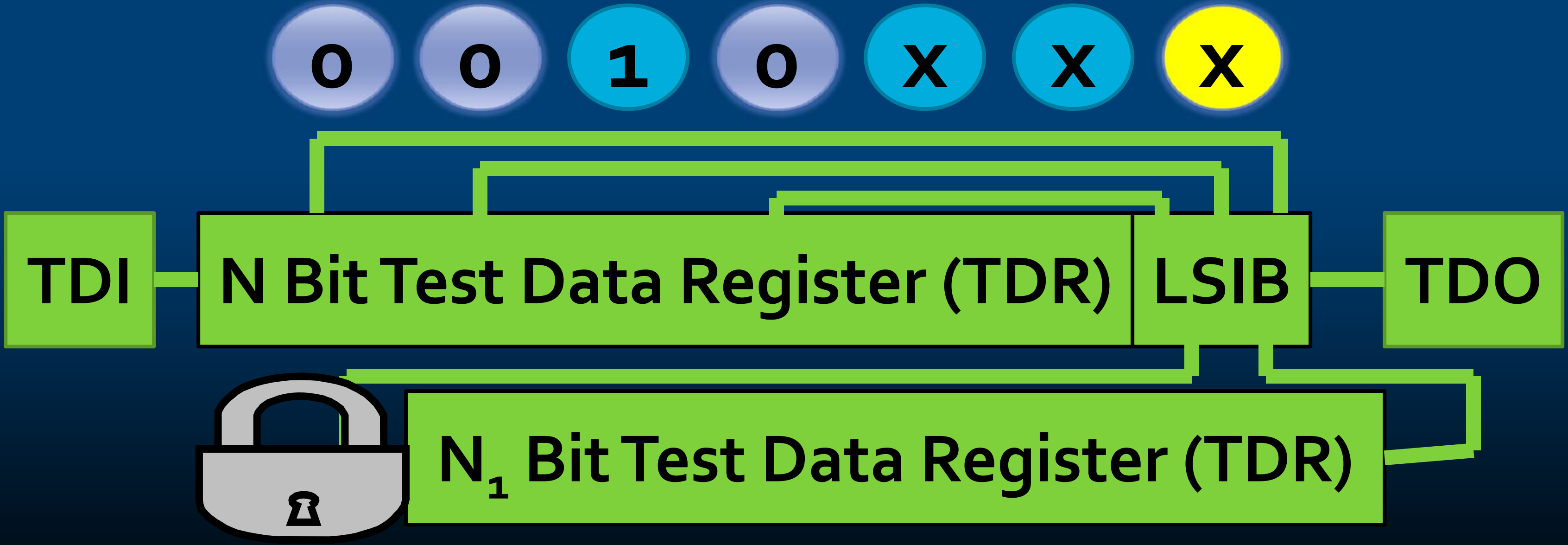
Distinguishing seq = 0100101

Shifting in guess.....

● LSIB

● Key

● Data


0
0
1
0
X
X
X
TDI
N Bit Test Data Register (TDR)
LSIB
TDO
N₁ Bit Test Data Register (TDR)

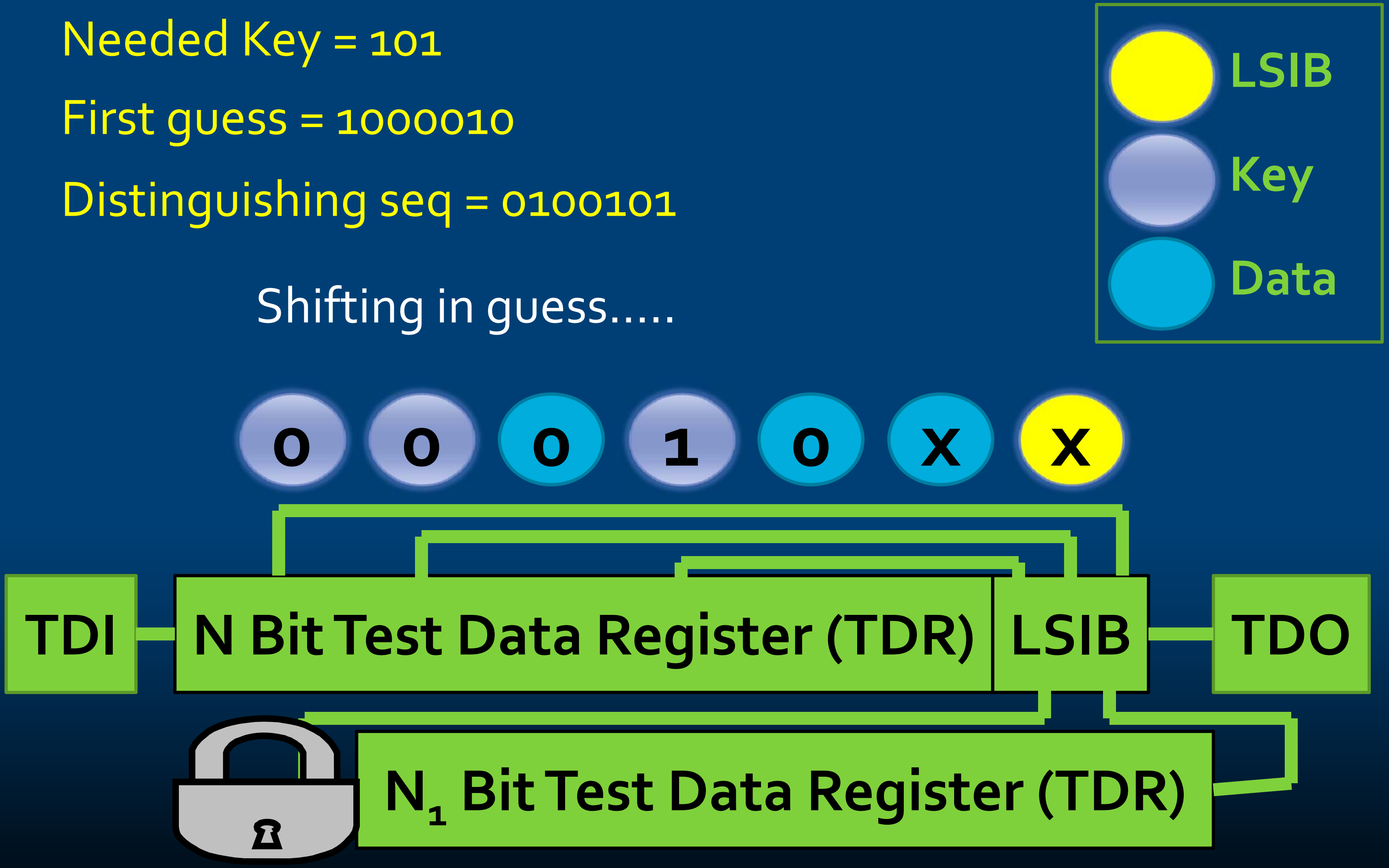
How Attackers Unlock LSIBs

Needed Key = 101

First guess = 1000010

Distinguishing seq = 0100101

Shifting in guess…..


LSIB
Key
Data
0
0
0
1
0
X
X
TDI
N Bit Test Data Register (TDR)
LSIB
TDO
N1 Bit Test Data Register (TDR)

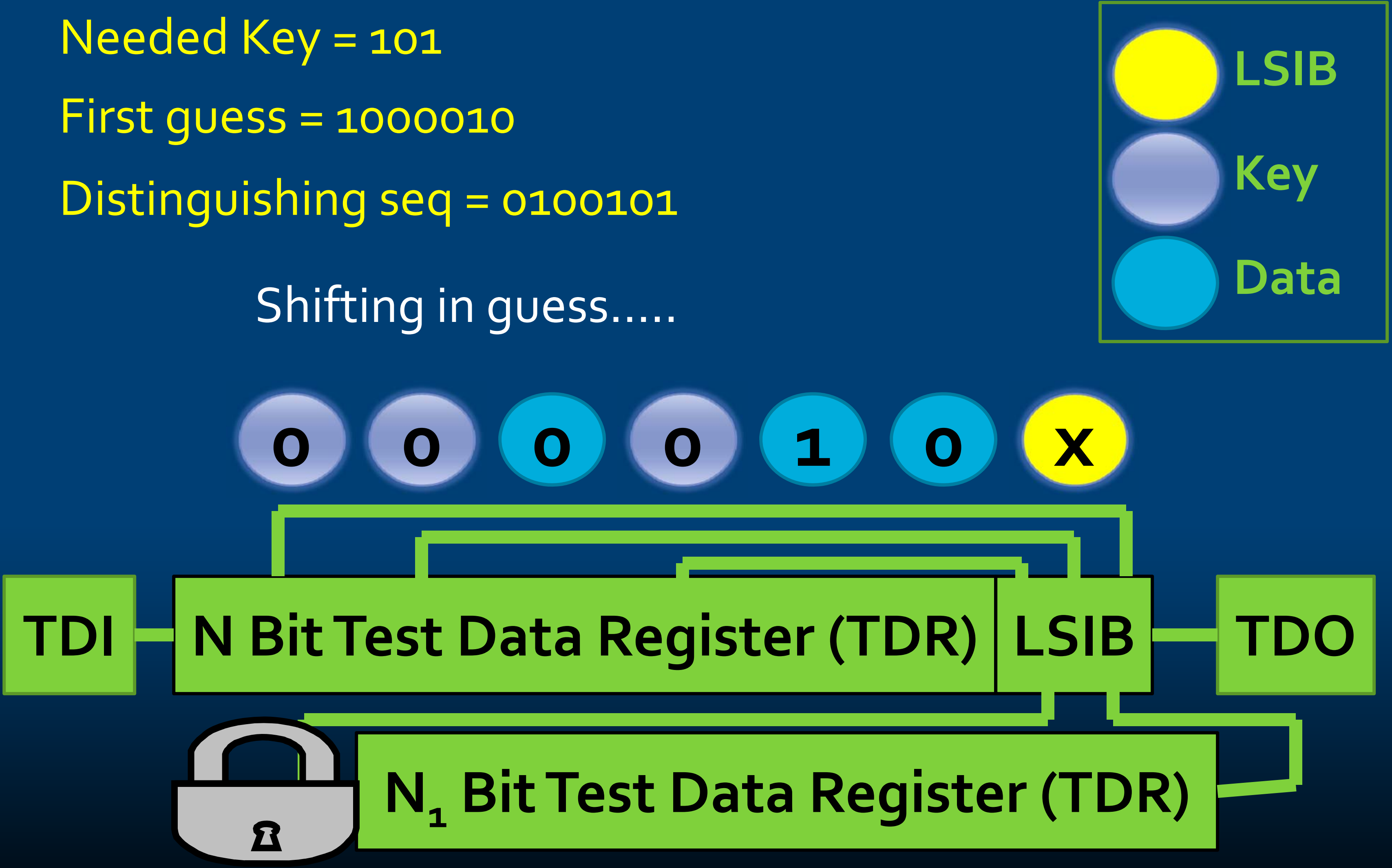
How Attackers Unlock LSIBs

Needed Key = 101

First guess = 1000010

Distinguishing seq = 0100101

Shifting in guess.....


LSIB
Key
Data
0
0
0
0
1
0
X
TDI
N Bit Test Data Register (TDR)
LSIB
TDO
N_1 Bit Test Data Register (TDR)

How Attackers Unlock LSIBs

Needed Key = 101

First guess = 1000010

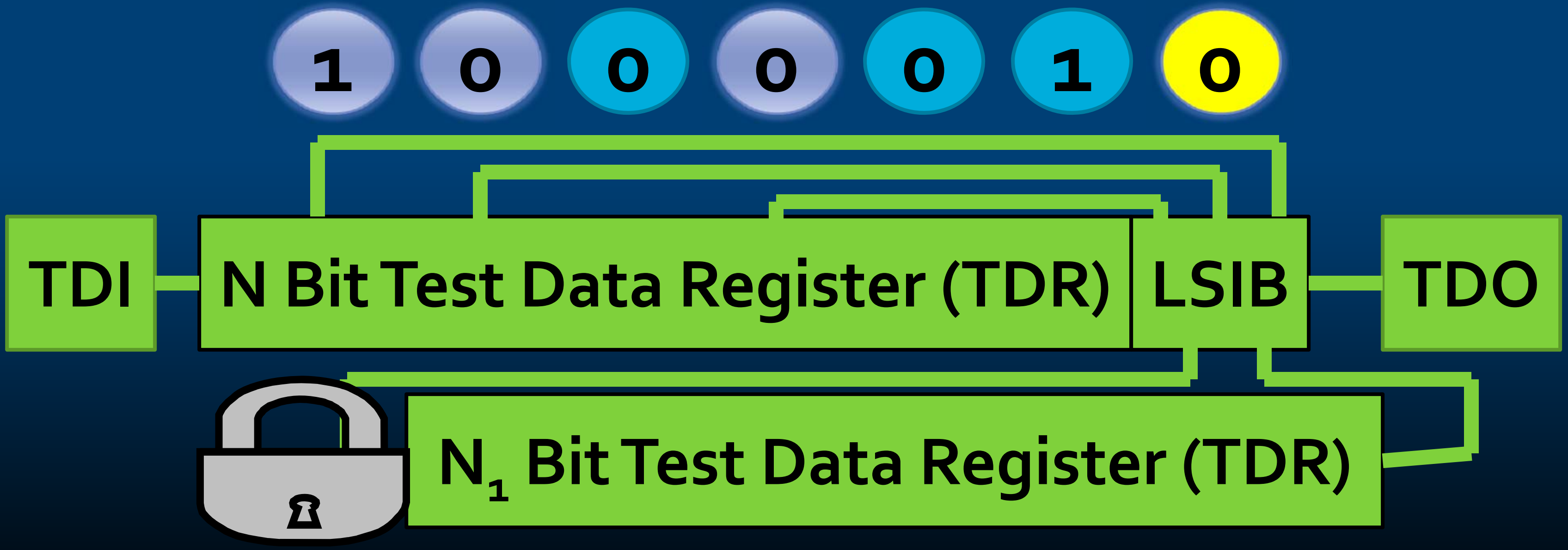
Distinguishing seq = 0100101

● LSIB

● Key

● Data

Shifting in guess…..


1
0
0
0
0
1
0
TDI
N Bit Test Data Register (TDR)
LSIB
TDO
N_1 Bit Test Data Register (TDR)

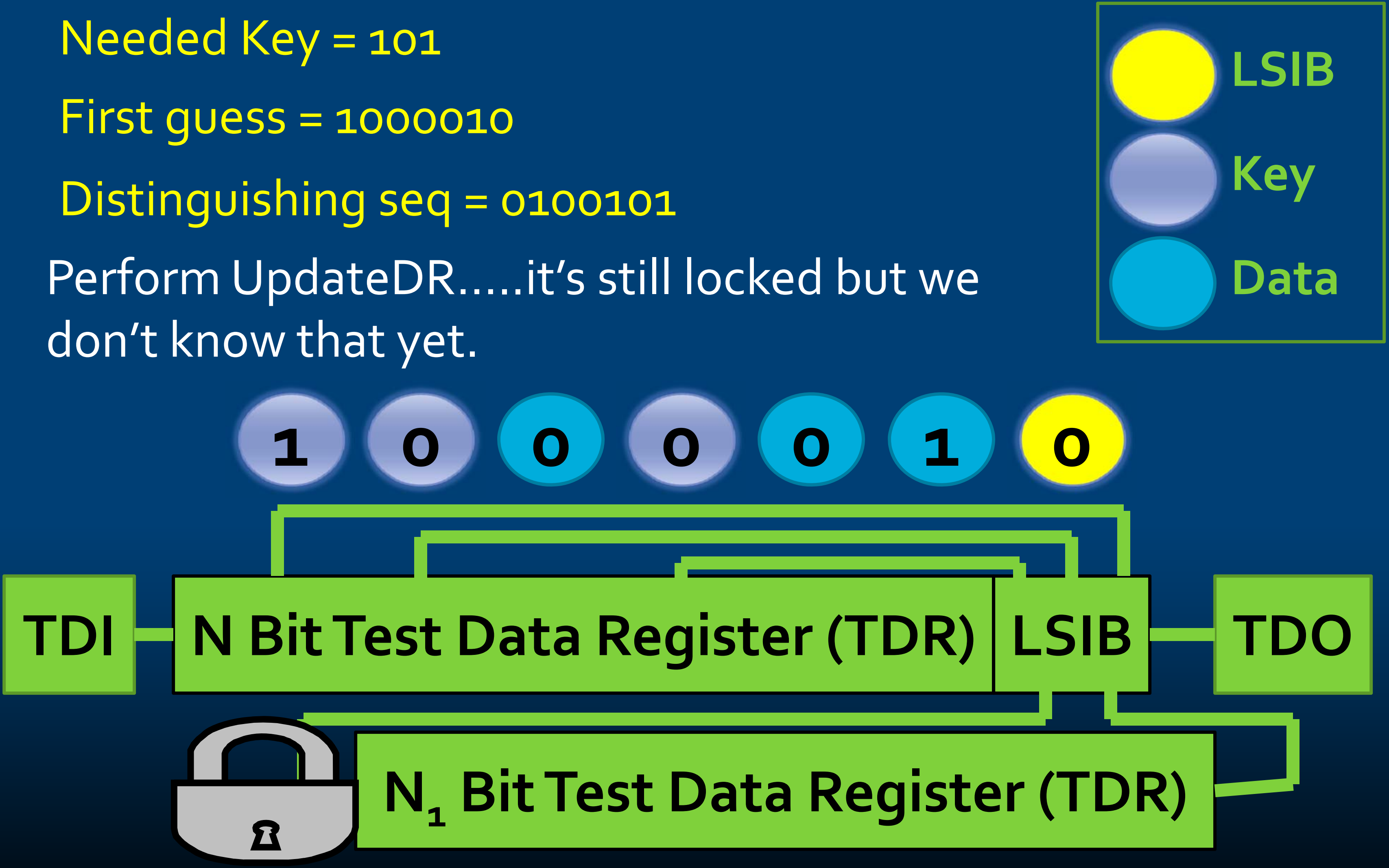
How Attackers Unlock LSIBs

Needed Key = 101

First guess = 1000010

Distinguishing seq = 0100101

Perform UpdateDR…..it's still locked but we don't know that yet.



LSIB
Key
Data
1
0
0
0
0
1
0
TDI
N Bit Test Data Register (TDR)
LSIB
TDO
N_1 Bit Test Data Register (TDR)

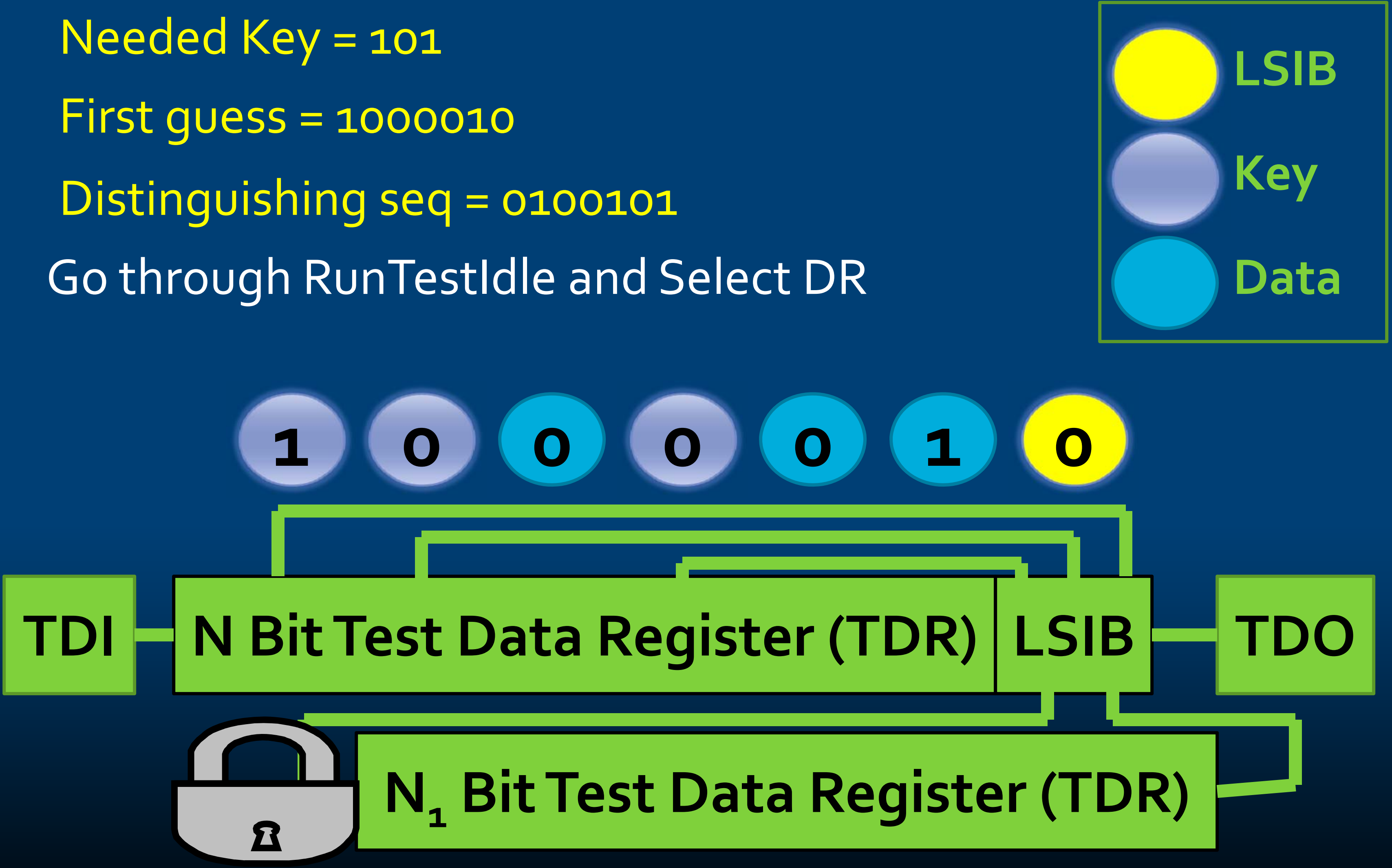
How Attackers Unlock LSIBs

Needed Key = 101

First guess = 1000010

Distinguishing seq = 0100101

Go through RunTestIdle and Select DR



LSIB
Key
Data
1
0
0
0
0
1
0
TDI
N Bit Test Data Register (TDR)
LSIB
TDO
N1 Bit Test Data Register (TDR)

How Attackers Unlock LSIBs

Needed Key = 101

First guess = 1000010

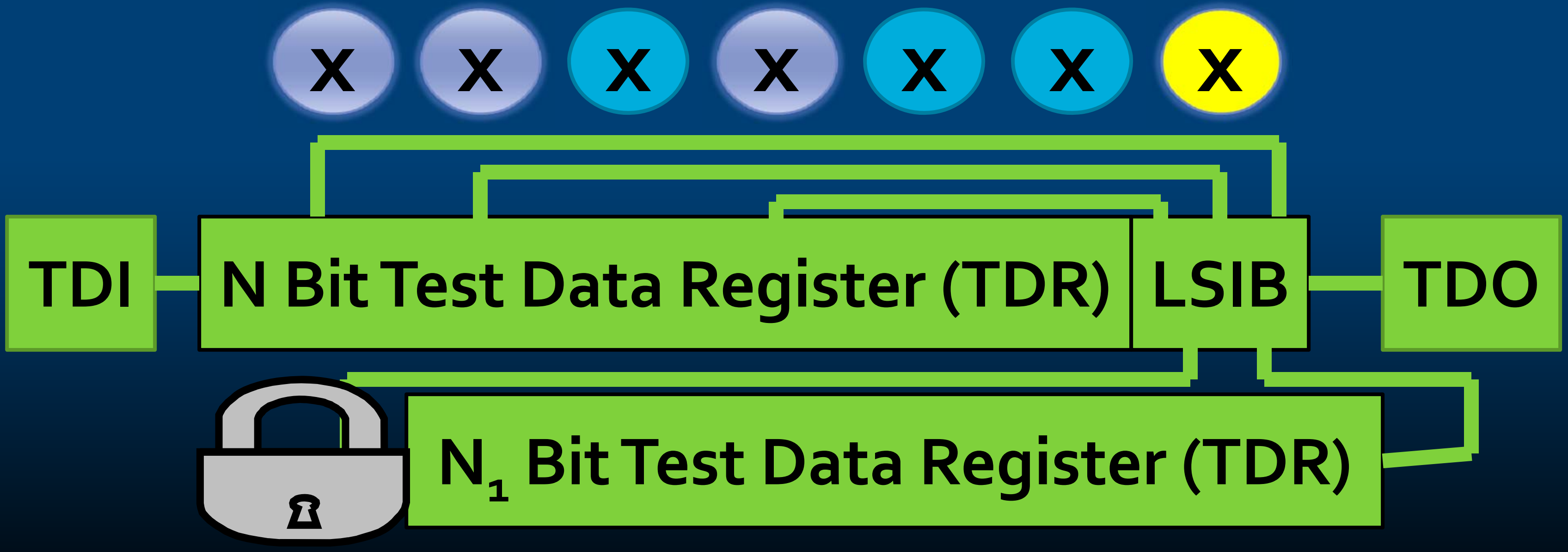
Distinguishing seq = 0100101

Go through CaptureDR & ShiftDR

🟡 LSIB

● Key

🔵 Data


x
x
x
x
x
x
x
TDI
N Bit Test Data Register (TDR)
LSIB
TDO
N₁ Bit Test Data Register (TDR)

How Attackers Unlock LSIBs

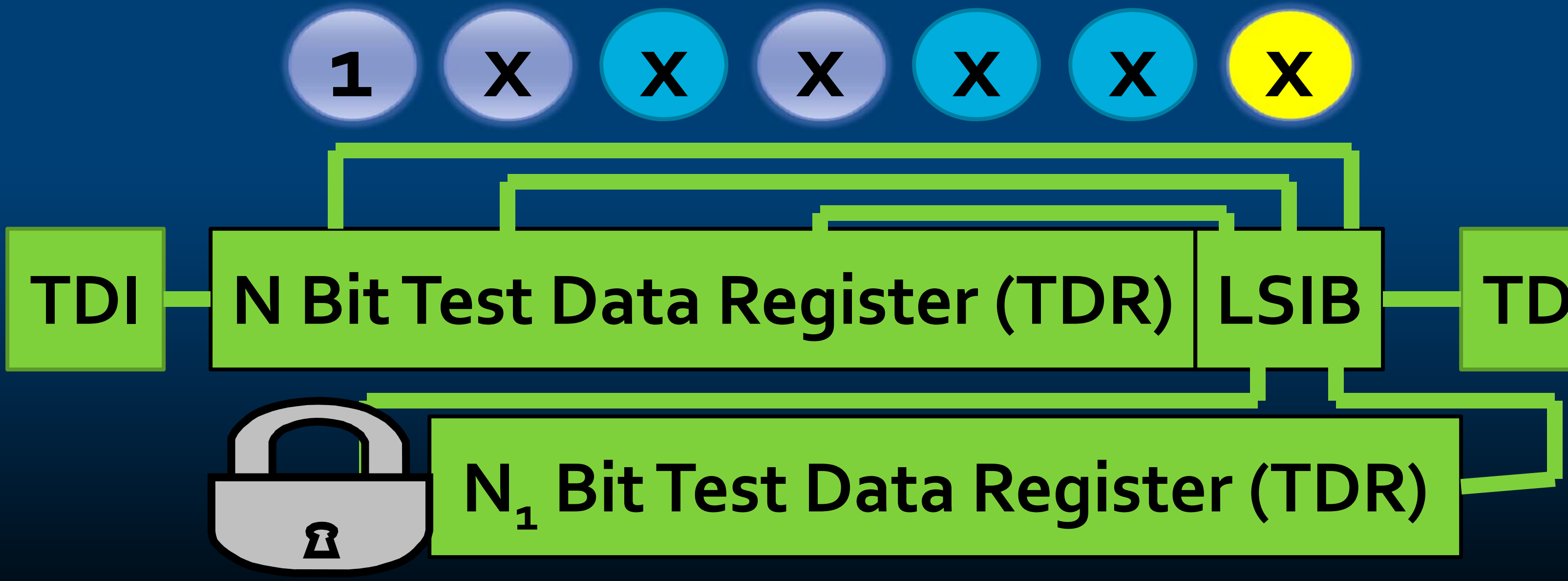
Needed Key = 101

First guess = 1000010

Distinguishing seq = 0100101

Start shifting in distinguishing sequence

🟡 LSIB
● Key
🔵 Data


1
x
x
x
x
x
x
TDI
N Bit Test Data Register (TDR)
LSIB
TDO
N_1 Bit Test Data Register (TDR)

How Attackers Unlock LSIBs

Needed Key = 101

First guess = 1000010

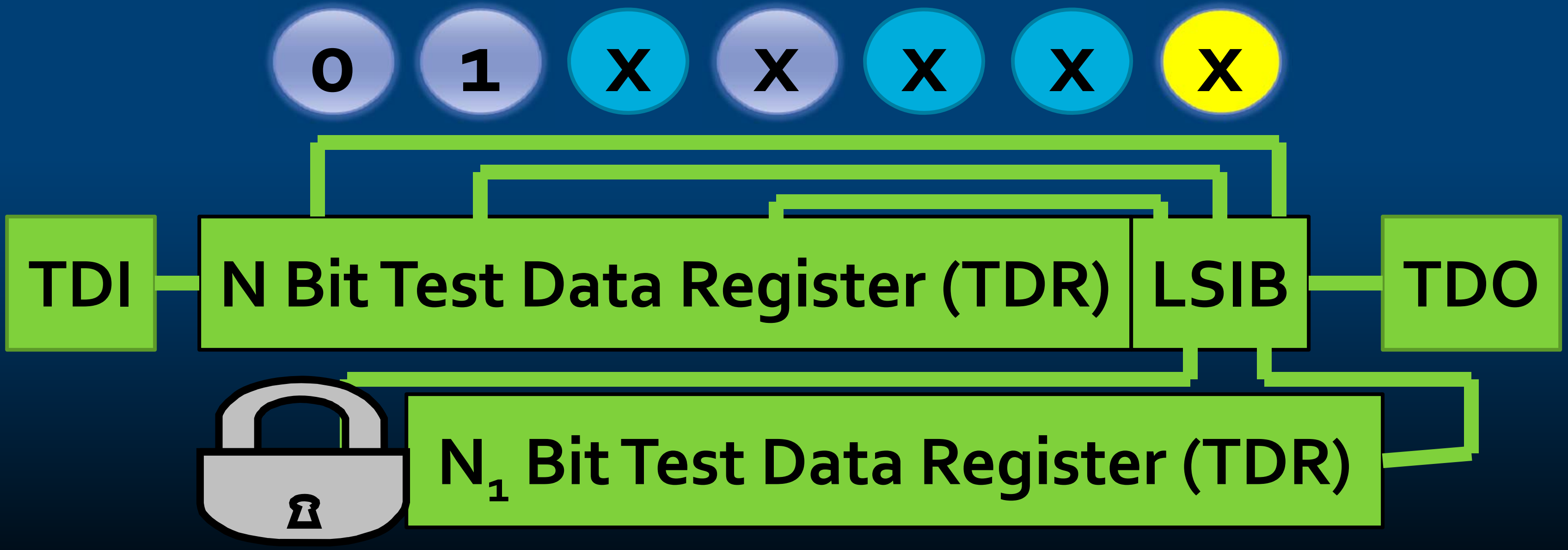
Distinguishing seq = 0100101

Start shifting in distinguishing sequence

● LSIB

● Key

● Data


0
1
X
X
X
X
X
TDI
N Bit Test Data Register (TDR)
LSIB
TDO
N_1 Bit Test Data Register (TDR)

How Attackers Unlock LSIBs

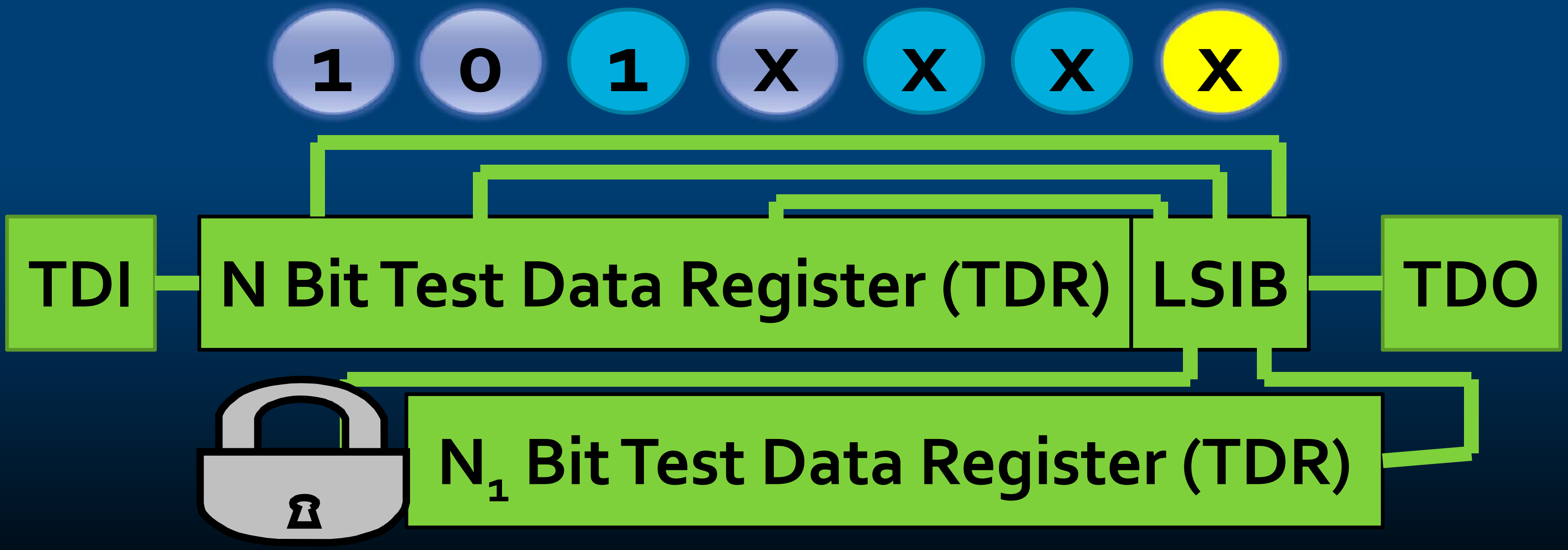
Needed Key = 101

First guess = 1000010

Distinguishing seq = 0100101

Start shifting in distinguishing sequence

- ● LSIB
- ● Key
- ● Data


1
0
1
X
X
X
X
TDI
N Bit Test Data Register (TDR)
LSIB
TDO
N_1 Bit Test Data Register (TDR)

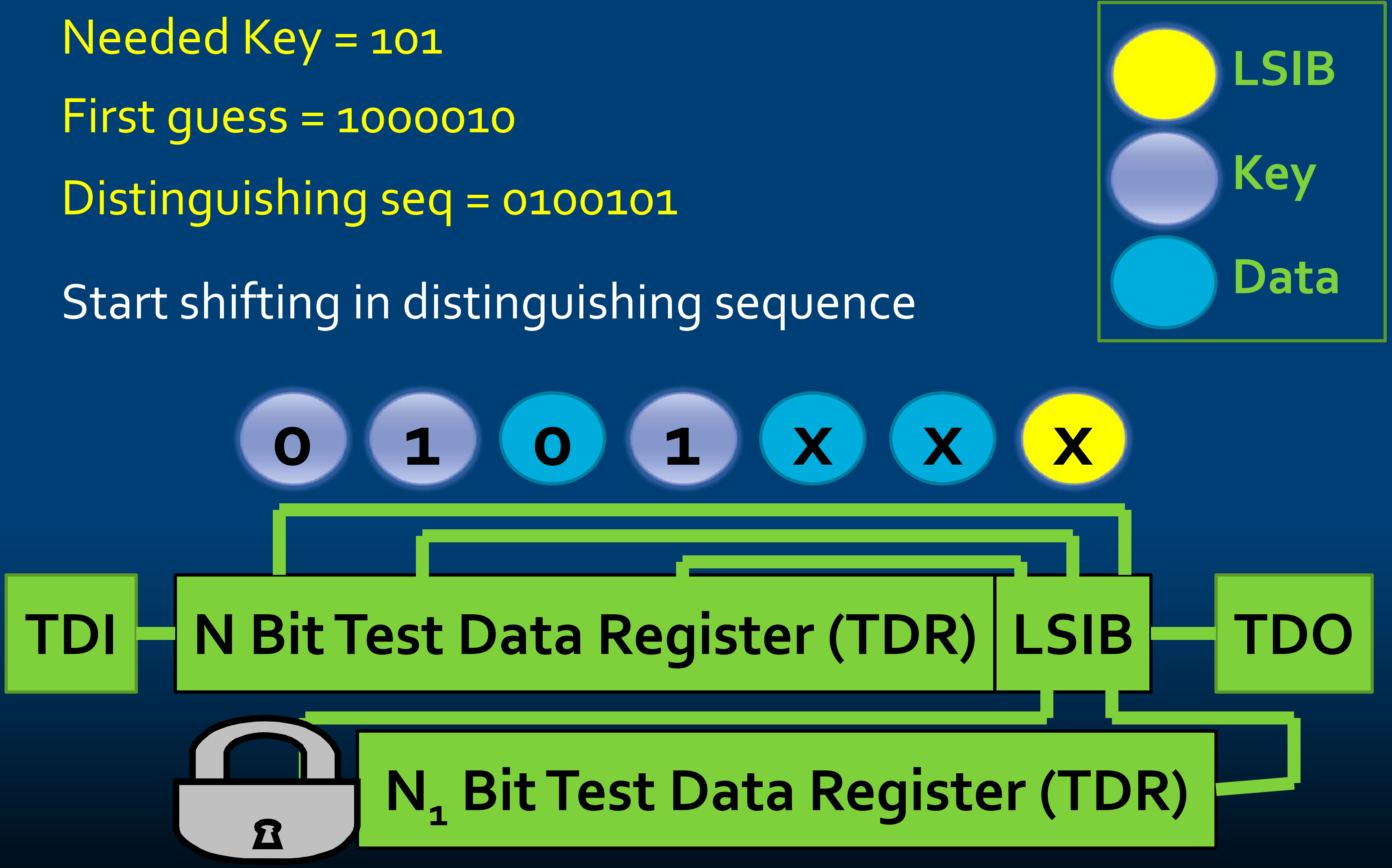
How Attackers Unlock LSIBs

Needed Key = 101

First guess = 1000010

Distinguishing seq = 0100101

Start shifting in distinguishing sequence



LSIB
Key
Data
0
1
0
1
X
X
X
TDI
N Bit Test Data Register (TDR)
LSIB
TDO
N1 Bit Test Data Register (TDR)

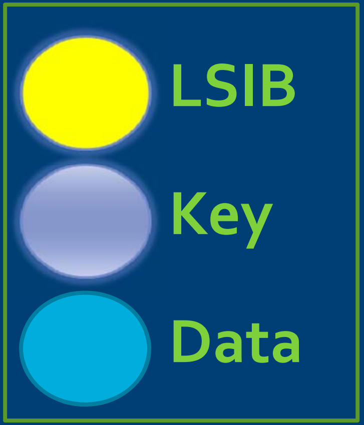
How Attackers Unlock LSIBs

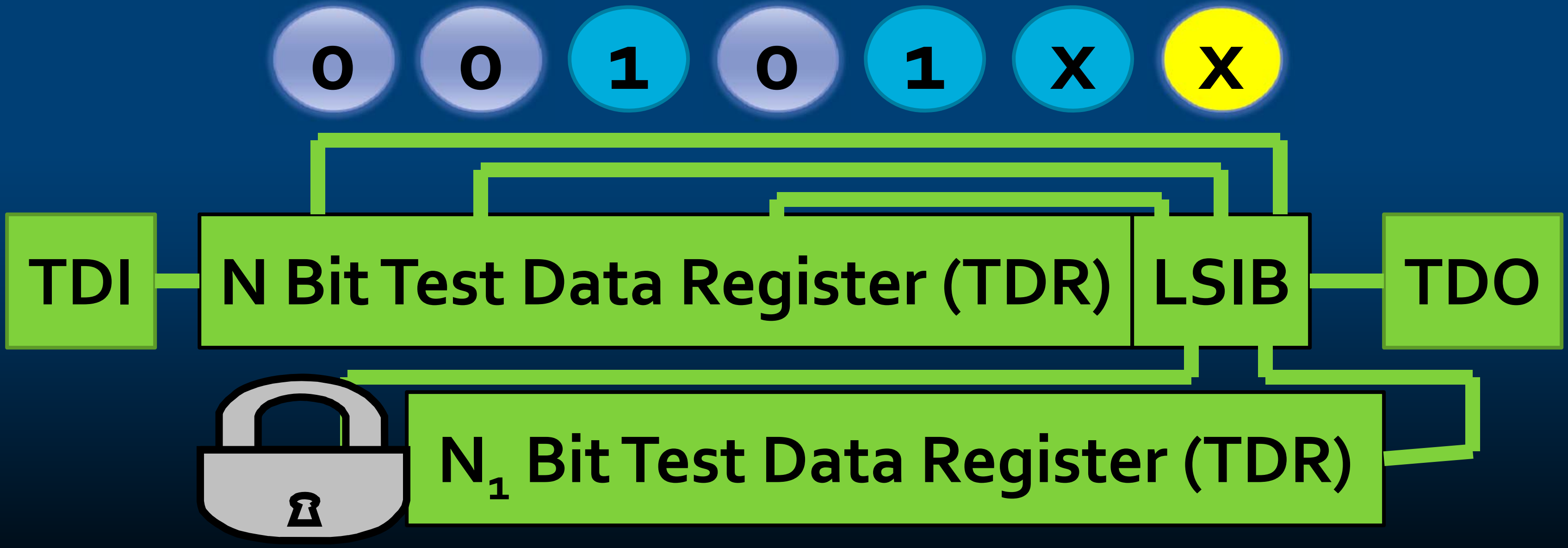
Needed Key = 101

First guess = 1000010

Distinguishing seq = 0100101

Start shifting in distinguishing sequence




0
0
1
0
1
X
X
TDI
N Bit Test Data Register (TDR)
LSIB
TDO
N₁ Bit Test Data Register (TDR)

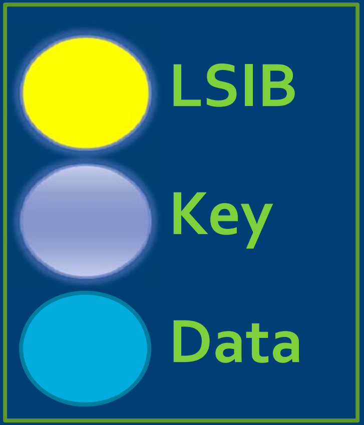
How Attackers Unlock LSIBs

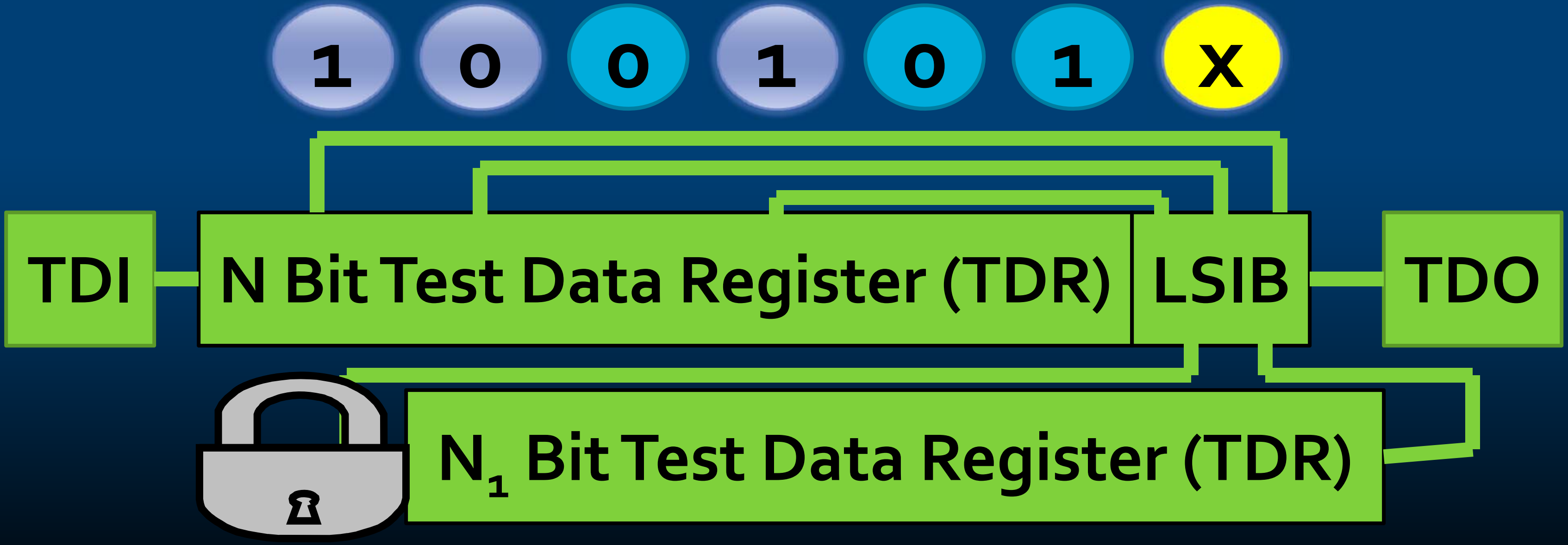
Needed Key = 101

First guess = 1000010

Distinguishing seq = 0100101

Start shifting in distinguishing sequence





1
0
0
1
0
1
X
TDI
N Bit Test Data Register (TDR)
LSIB
TDO
N_1 Bit Test Data Register (TDR)

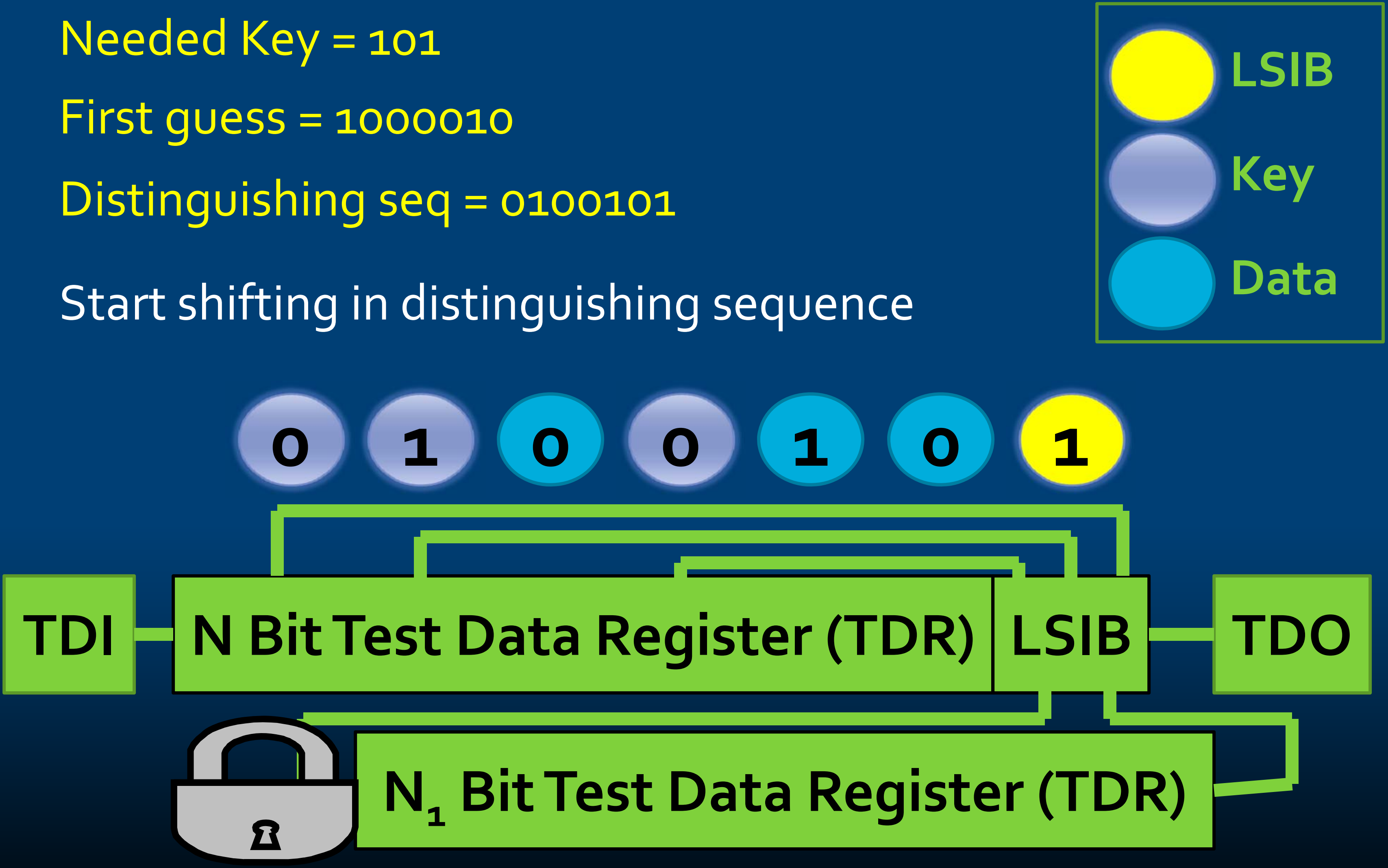
How Attackers Unlock LSIBs

Needed Key = 101

First guess = 1000010

Distinguishing seq = 0100101

Start shifting in distinguishing sequence


LSIB
Key
Data
0
1
0
0
1
0
1
TDI
N Bit Test Data Register (TDR)
LSIB
TDO
N_1 Bit Test Data Register (TDR)

How Attackers Unlock LSIBs

Needed Key = 101

Second guess = 1011100

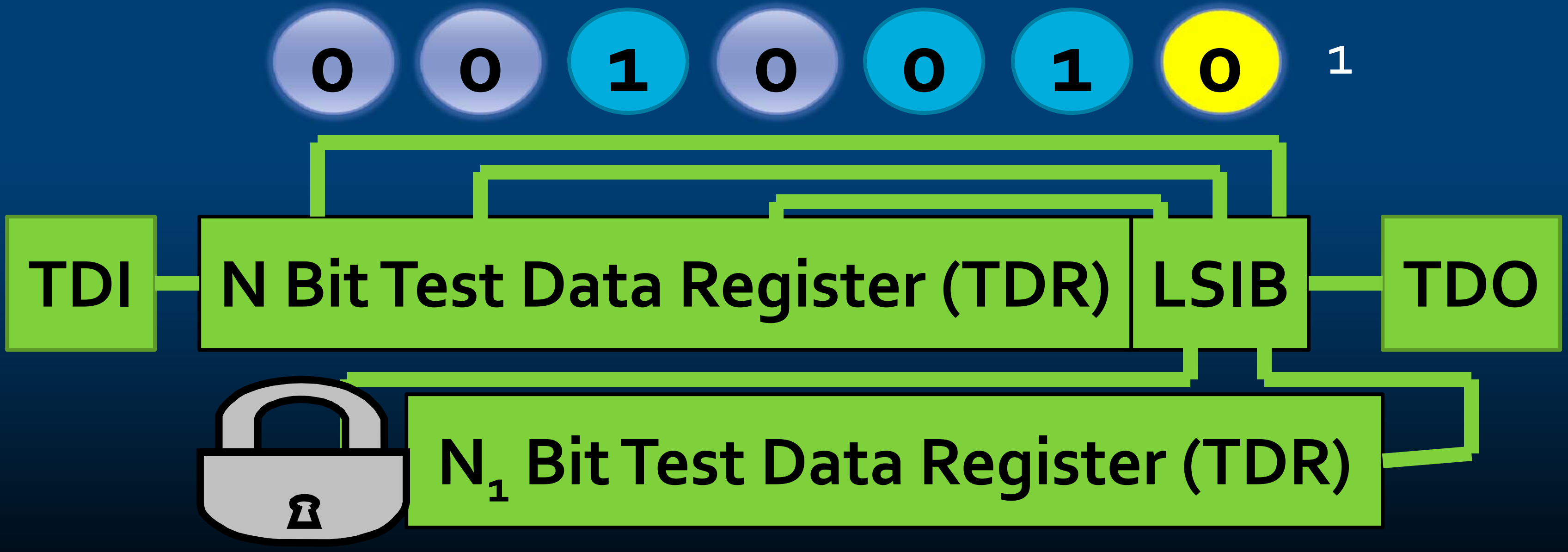
Distinguishing seq = 0100101

Start shifting in second guess

● LSIB

● Key

● Data


0
0
1
0
0
1
0
1
TDI
N Bit Test Data Register (TDR)
LSIB
TDO
N_1 Bit Test Data Register (TDR)

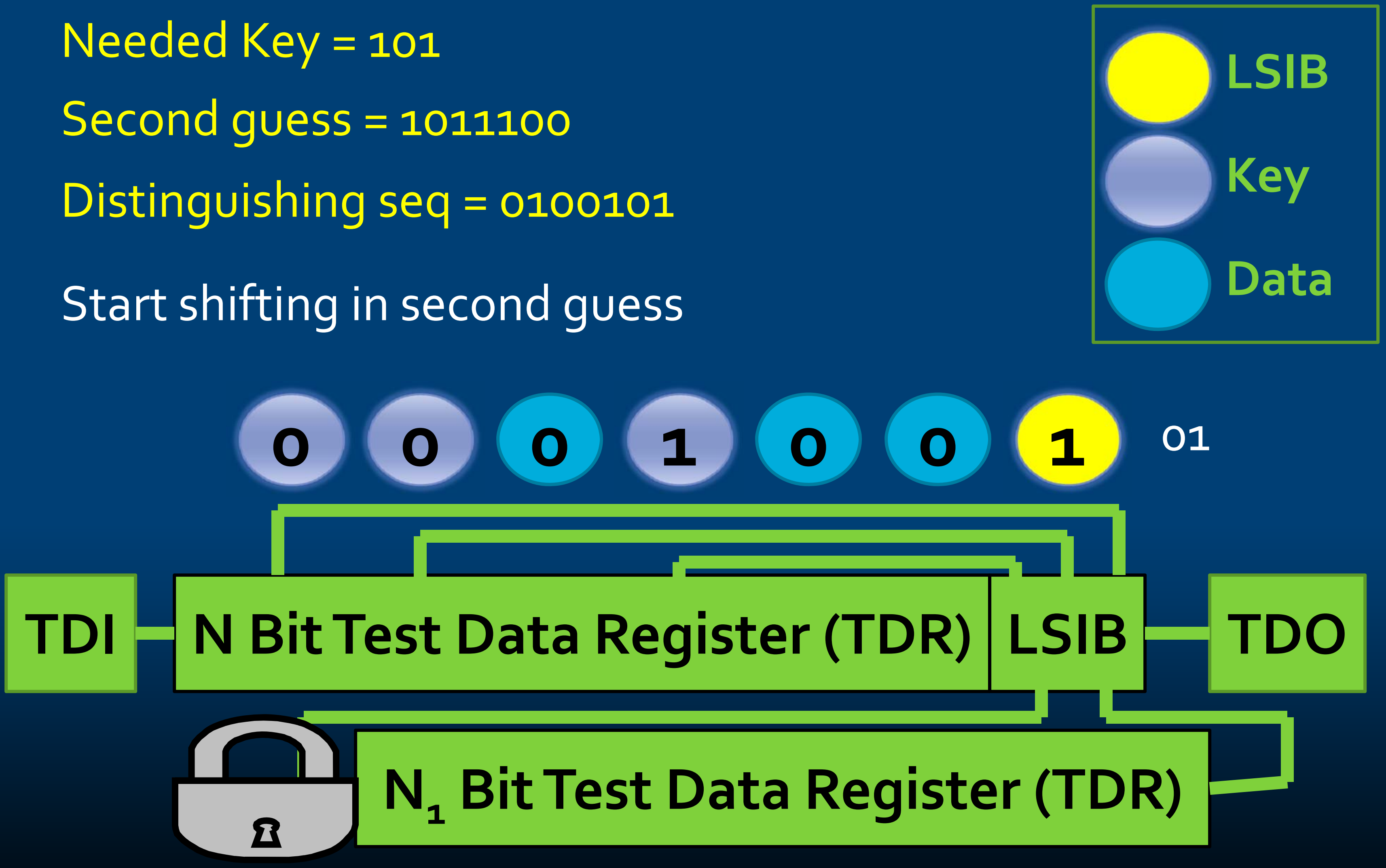
How Attackers Unlock LSIBs

Needed Key = 101

Second guess = 1011100

Distinguishing seq = 0100101

Start shifting in second guess


LSIB
Key
Data
0
0
0
1
0
0
1
01
TDI
N Bit Test Data Register (TDR)
LSIB
TDO
N1 Bit Test Data Register (TDR)

How Attackers Unlock LSIBs

Needed Key = 101

Second guess = 1011100

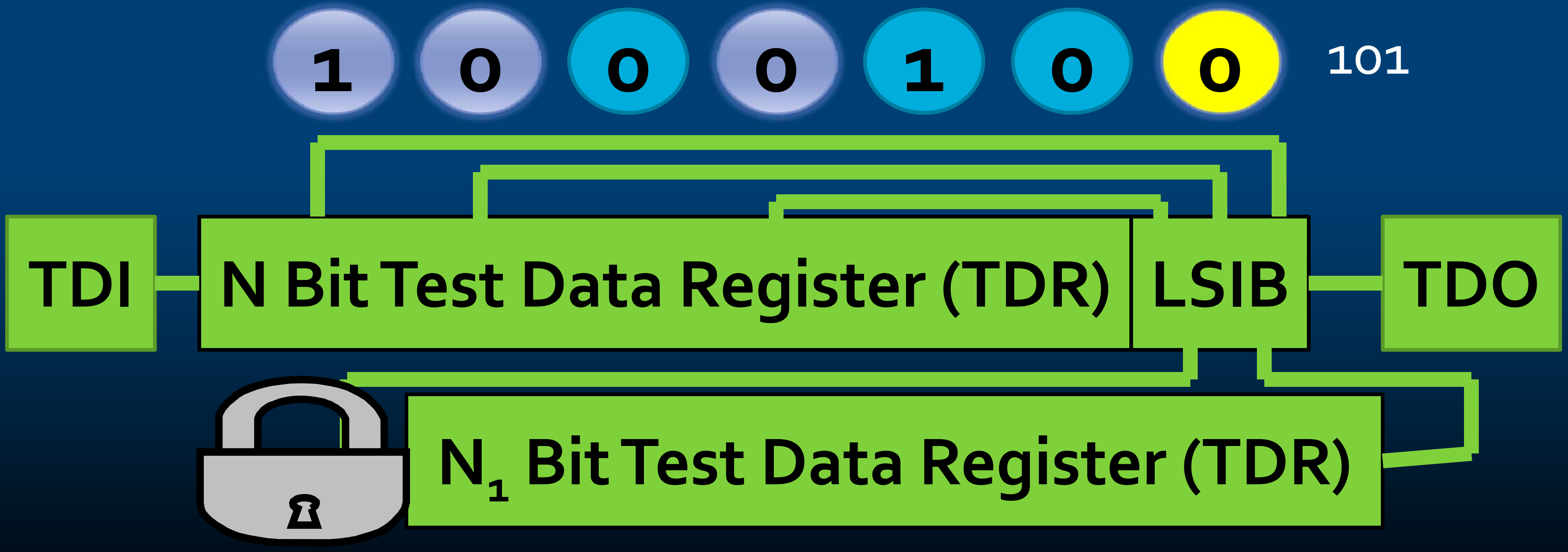
Distinguishing seq = 0100101

Start shifting in second guess

● LSIB

● Key

● Data


1
0
0
0
1
0
0
101
TDI
N Bit Test Data Register (TDR)
LSIB
TDO
N₁ Bit Test Data Register (TDR)

How Attackers Unlock LSIBs

Needed Key = 101

Second guess = 1011100

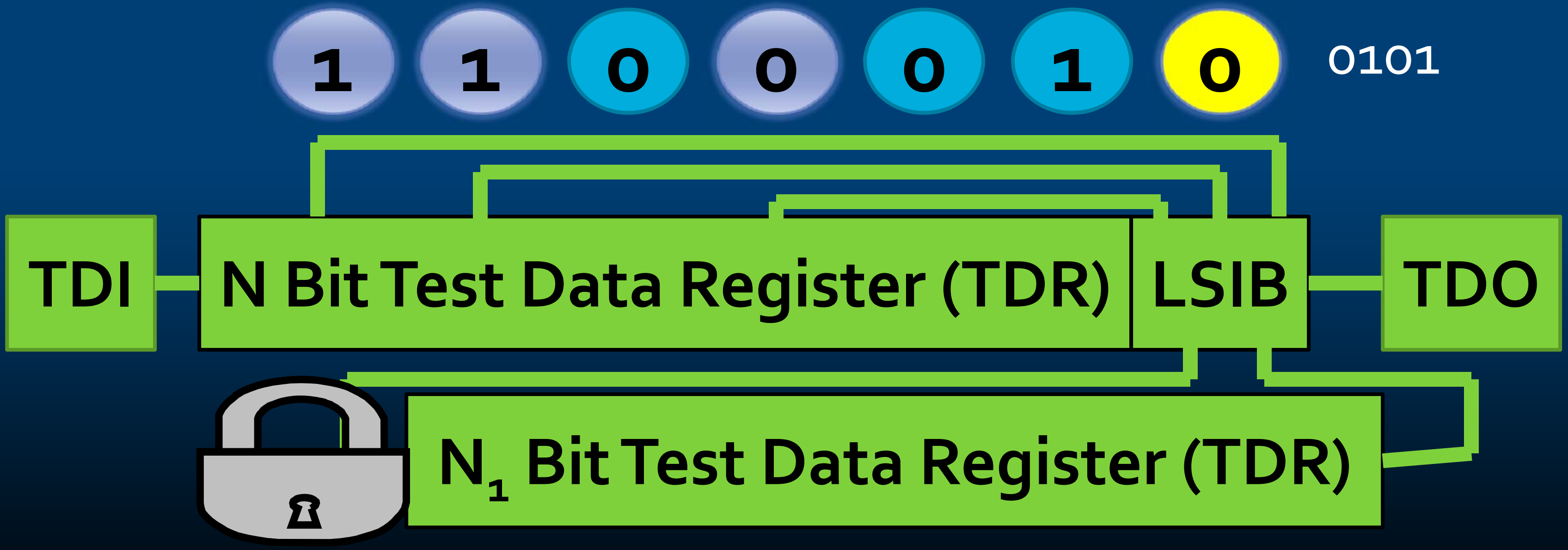
Distinguishing seq = 0100101

Start shifting in second guess

● LSIB

● Key

● Data


1
1
0
0
0
1
0
0101
TDI
N Bit Test Data Register (TDR)
LSIB
TDO
N₁ Bit Test Data Register (TDR)

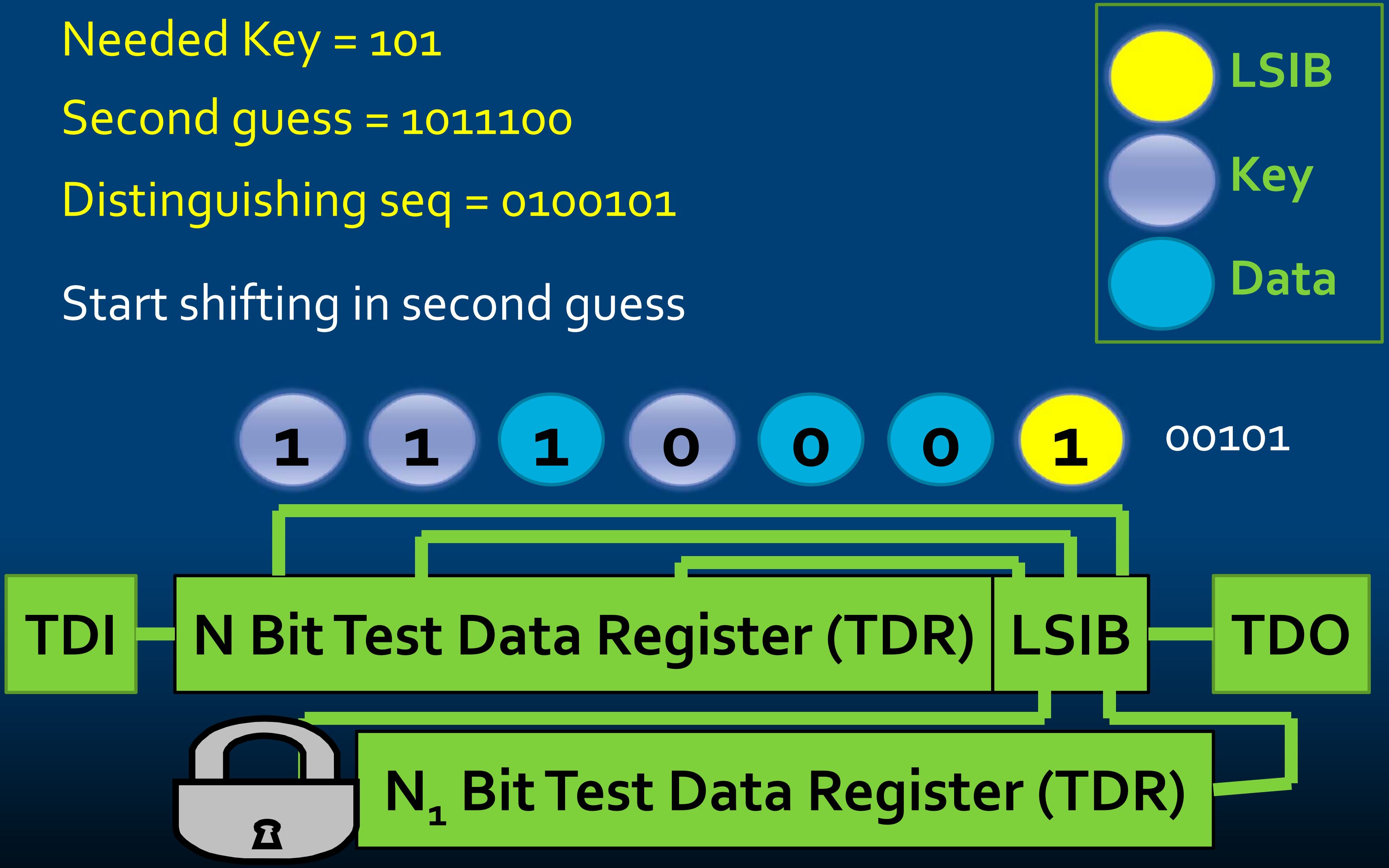
How Attackers Unlock LSIBs

Needed Key = 101

Second guess = 1011100

Distinguishing seq = 0100101

Start shifting in second guess



LSIB
Key
Data
1
1
1
0
0
0
1
00101
TDI
N Bit Test Data Register (TDR)
LSIB
TDO
N₁ Bit Test Data Register (TDR)

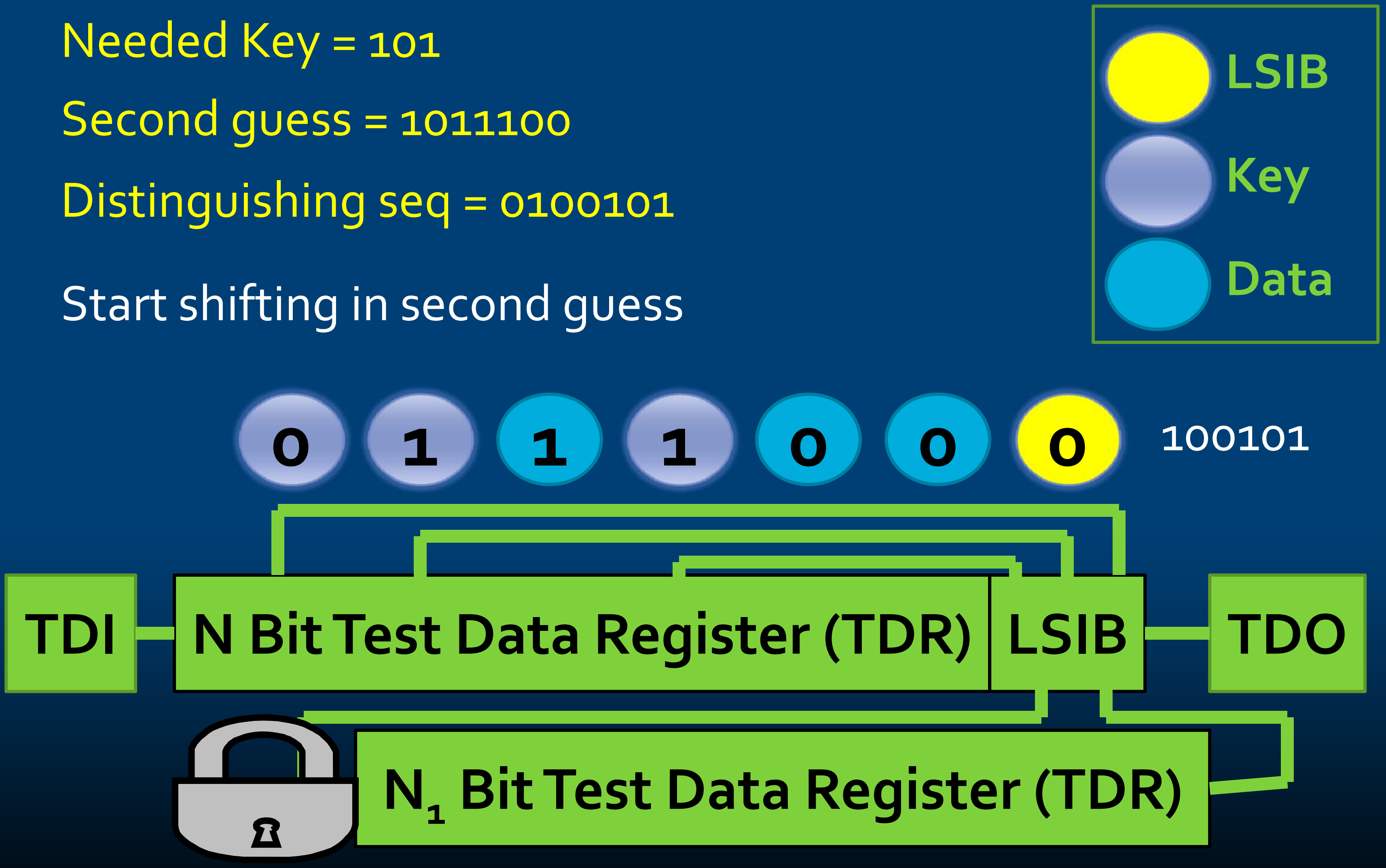
How Attackers Unlock LSIBs

Needed Key = 101

Second guess = 1011100

Distinguishing seq = 0100101

Start shifting in second guess


LSIB
Key
Data
0
1
1
1
0
0
0
100101
TDI
N Bit Test Data Register (TDR)
LSIB
TDO
N_1 Bit Test Data Register (TDR)

How Attackers Unlock LSIBs

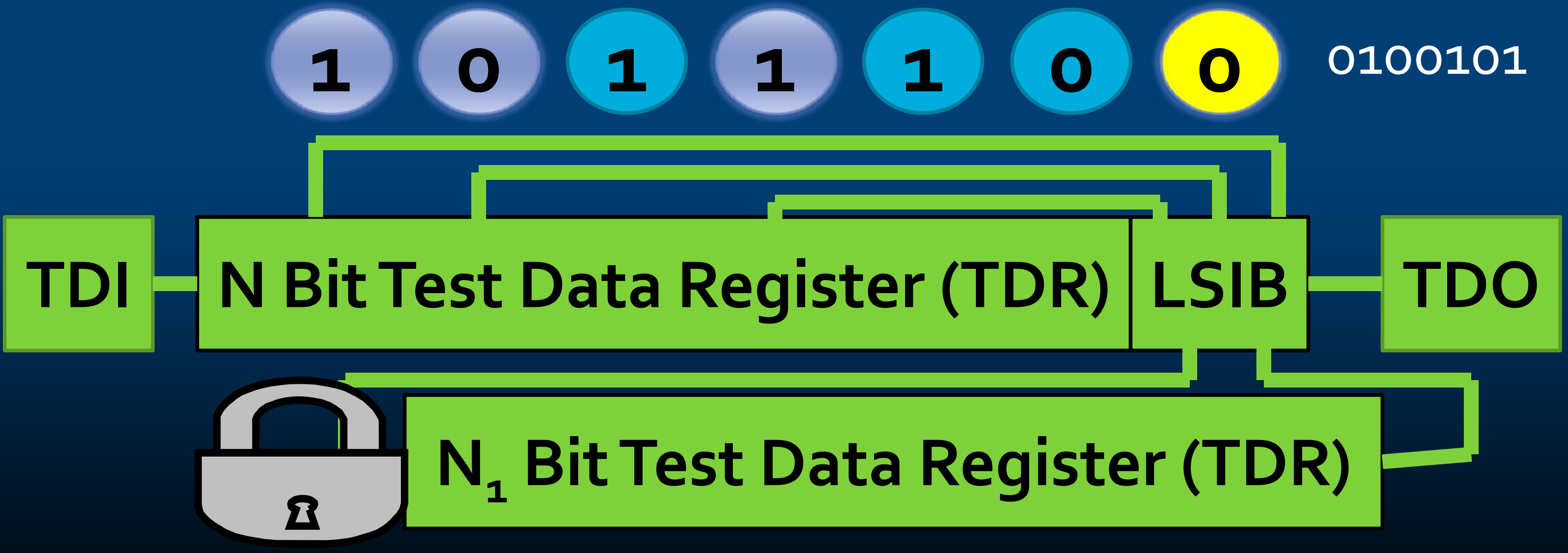
Needed Key = 101

Second guess = 1011100

Distinguishing seq = 0100101

Distinguishing sequence out when expected…
Second guess is in chain….

● LSIB
● Key
● Data


1
0
1
1
1
0
0
0100101
TDI
N Bit Test Data Register (TDR)
LSIB
TDO
N_1 Bit Test Data Register (TDR)

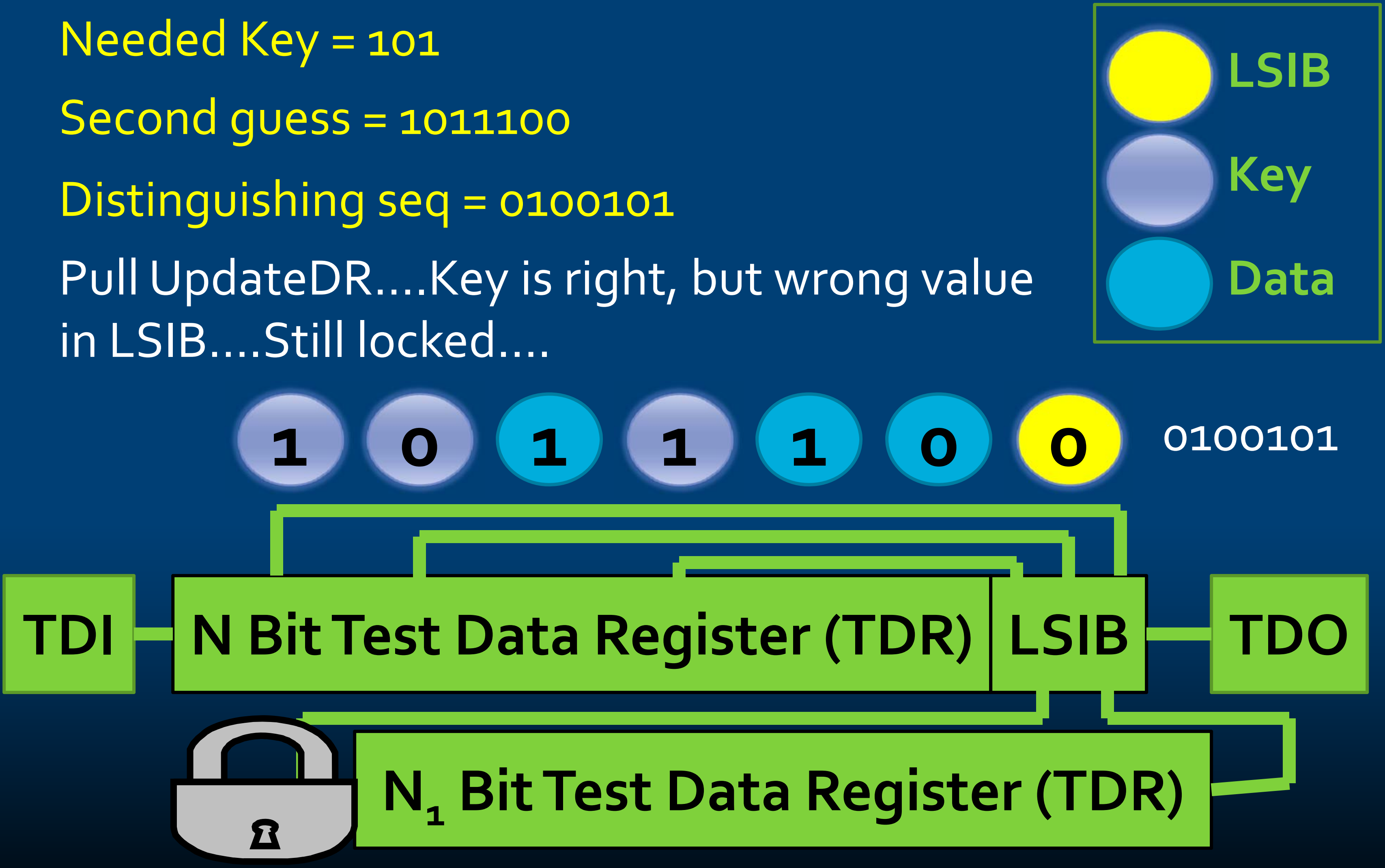
How Attackers Unlock LSIBs

Needed Key = 101

Second guess = 1011100

Distinguishing seq = 0100101

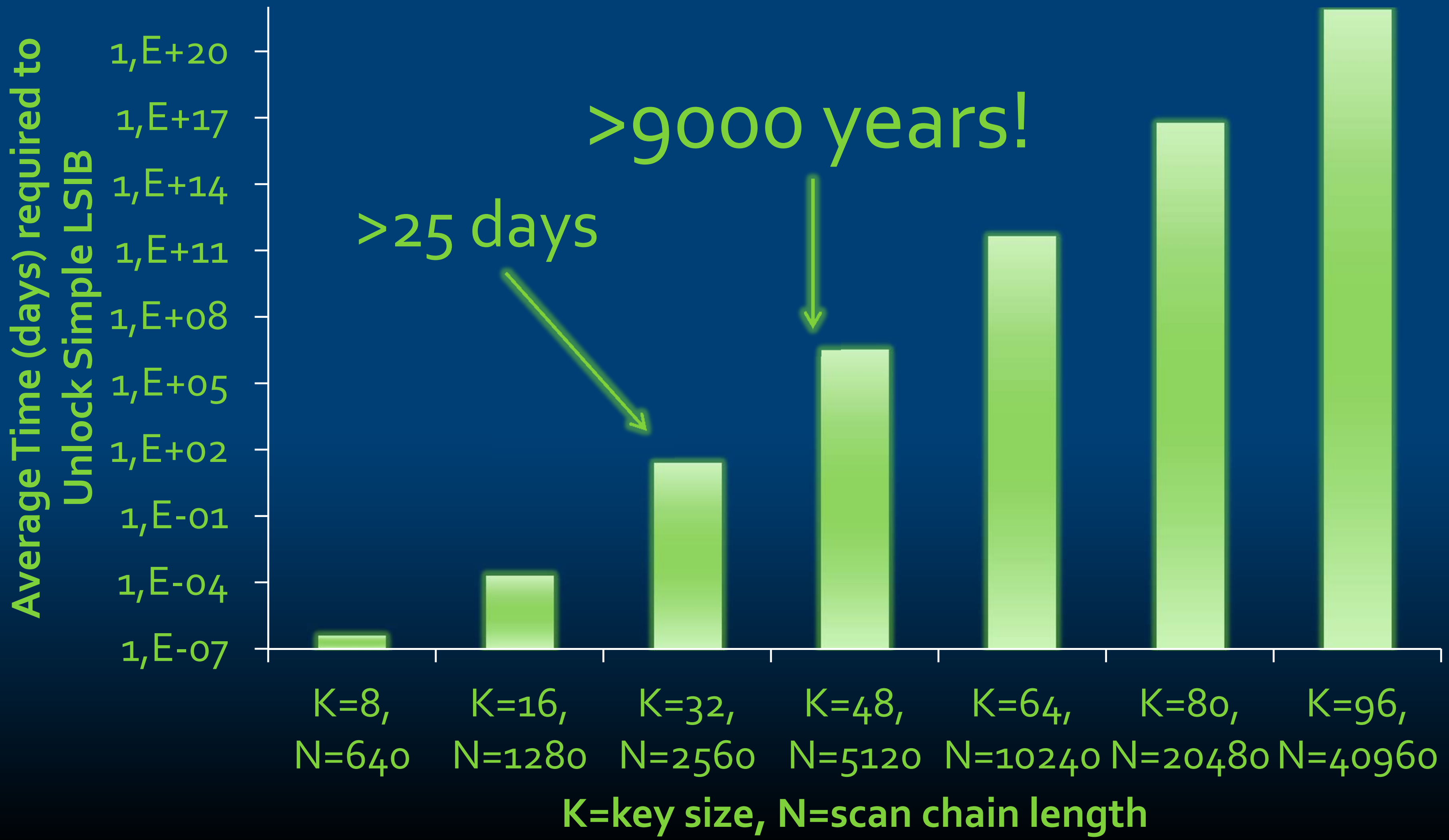
Pull UpdateDR….Key is right, but wrong value in LSIB….Still locked….


LSIB
Key
Data
1
0
1
1
1
0
0
0100101
TDI
N Bit Test Data Register (TDR)
LSIB
TDO
N1 Bit Test Data Register (TDR)

How long until the attacker successful?

- Number of possible combinations for k bits in the key and 1 value LSIB = $2^{(k+1)}$
- Probability of guessing the correct combination = $1/2^{(k+1)}$
- Expected number of tries before the correct guess is selected = $2^{(k+1)}$
- Cost of a guess = 5+n+d clock cycles
- Total expected clock cycles to open LSIB with random guesses = $(5+n+d) * 2^{(k+1)}$

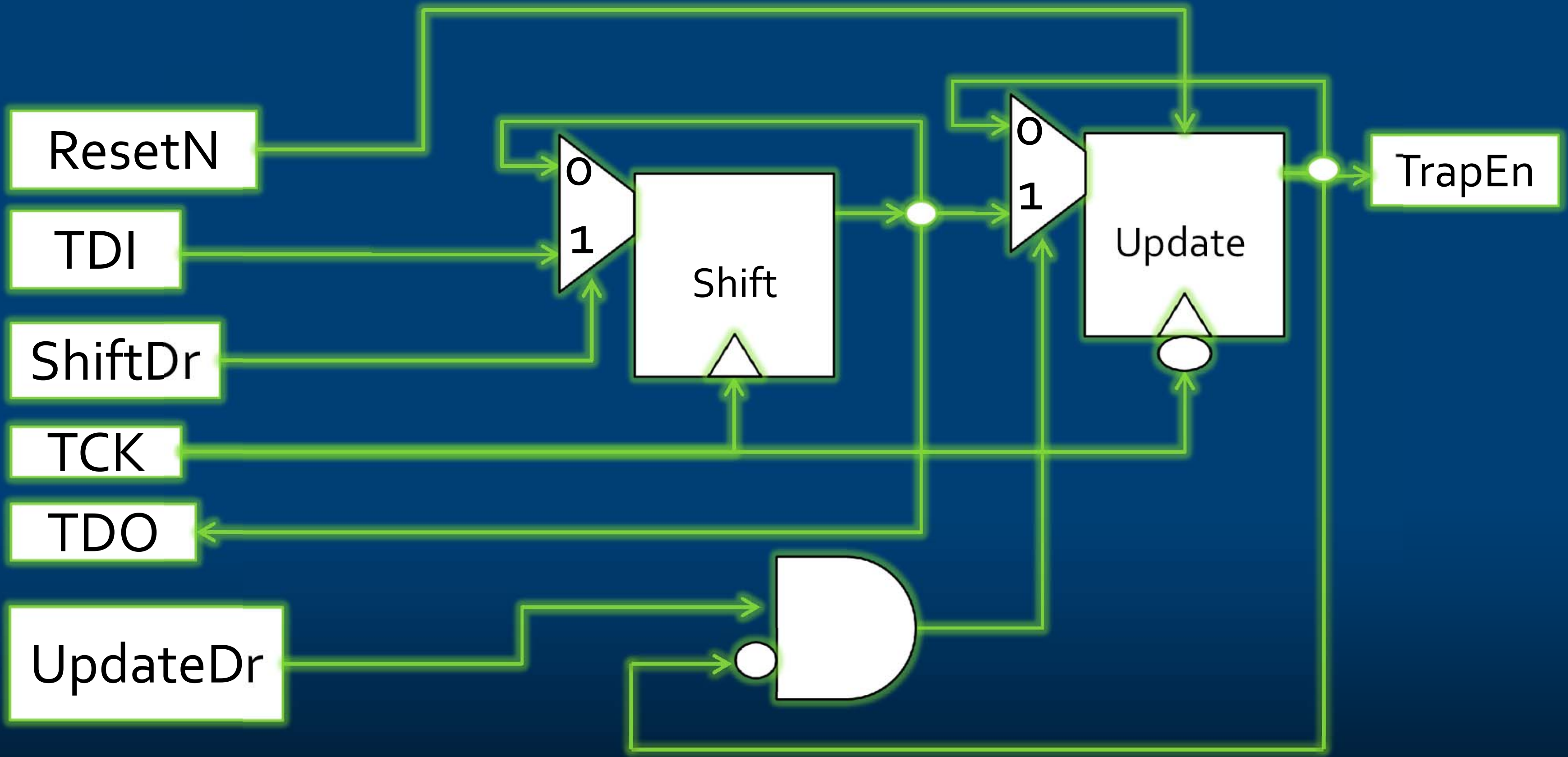
How long does it take to open an LSIB (assuming shifting at 10 MHz & d=25)?



Of course, an attacker could be lucky…and guess correctly on the first couple of tries…can we make it harder?

Yes!!! We can set traps and use hierarchical LSIBs architectures….

TRAPs—Once you fall in, you can’t get out…



- Trap is sprung if the wrong value (1) present on an updateDR
- TrapEn can keep an LSIB from opening or a key-bit from updating
- Can be designed to only reset on global or local reset

How does the attacker make a guess with TRAPs present?

Update DR	Update DR needed to try to open LSIB: 1 cycle.
Run Test Idle	1 cycle.
Select DR	1 cycle.
Capture DR	1 cycle.
Shift DR	1 cycle.
Shift Cycles	N+d cycles to check of chain length.

Reset	Disable Traps: 1 cycle.
Reload IR	Depends on RESET Behavior and IR size.
Run Test Idle	1 cycle.
Select DR	1 cycle.
Capture DR	1 cycle.
Shift DR	1 cycle.
Shift Cycles	New pattern shifted in: N cycles

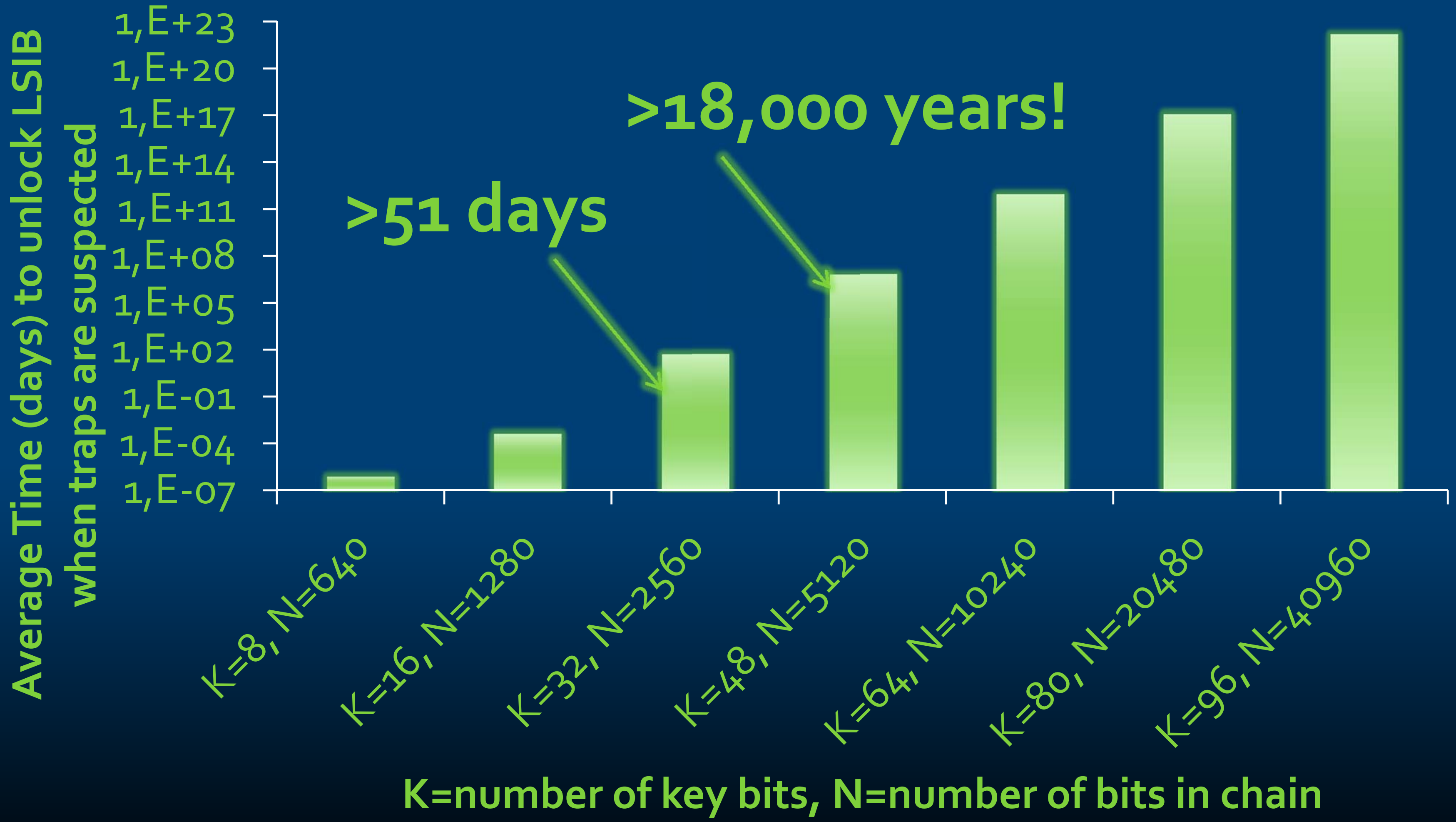
How does the attacker make a guess with TRAPs present?

Update DR	Update DR needed to try to open LSIB: 1 cycle.
Run Test Idle	1 cycle.
Select DR	
Capture DR	1 cycle.
Shift DR	1 cycle.
Shift Cycles	N+d cycles to check of chain length.

Reset	Disable Traps: 1 cycle.
Reload IR	Depends on RESET Behavior and IR size.
Capture DR	1 cycle.
Shift DR	1 cycle.
Shift Cycles	New pattern shifted in: N cycles

Cost of a guess if attacker thinks Traps may be Present: >= 10 + 2n + d

How long does it take to open an LSIB with traps?



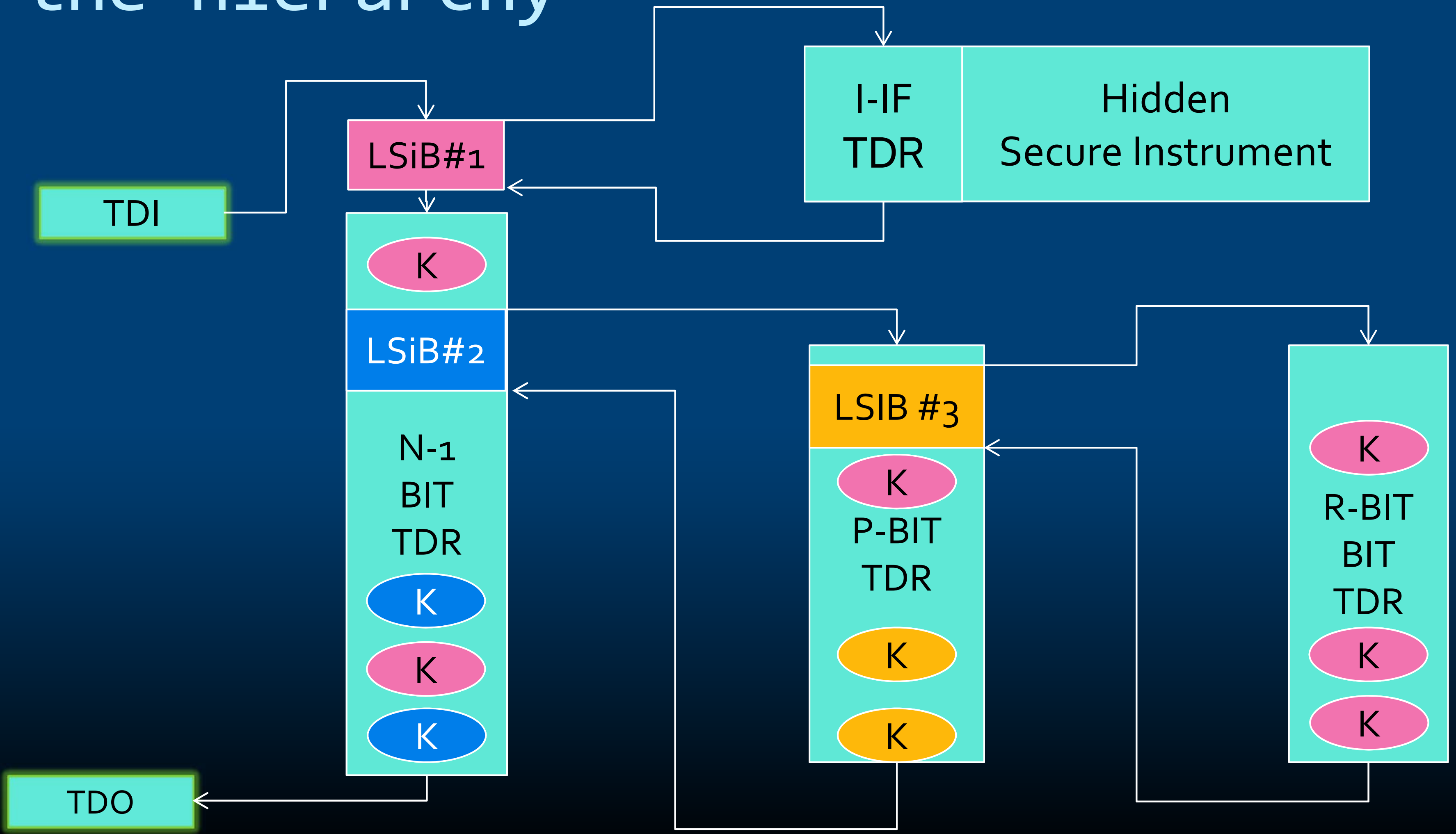
>18,000 years!

>51 days

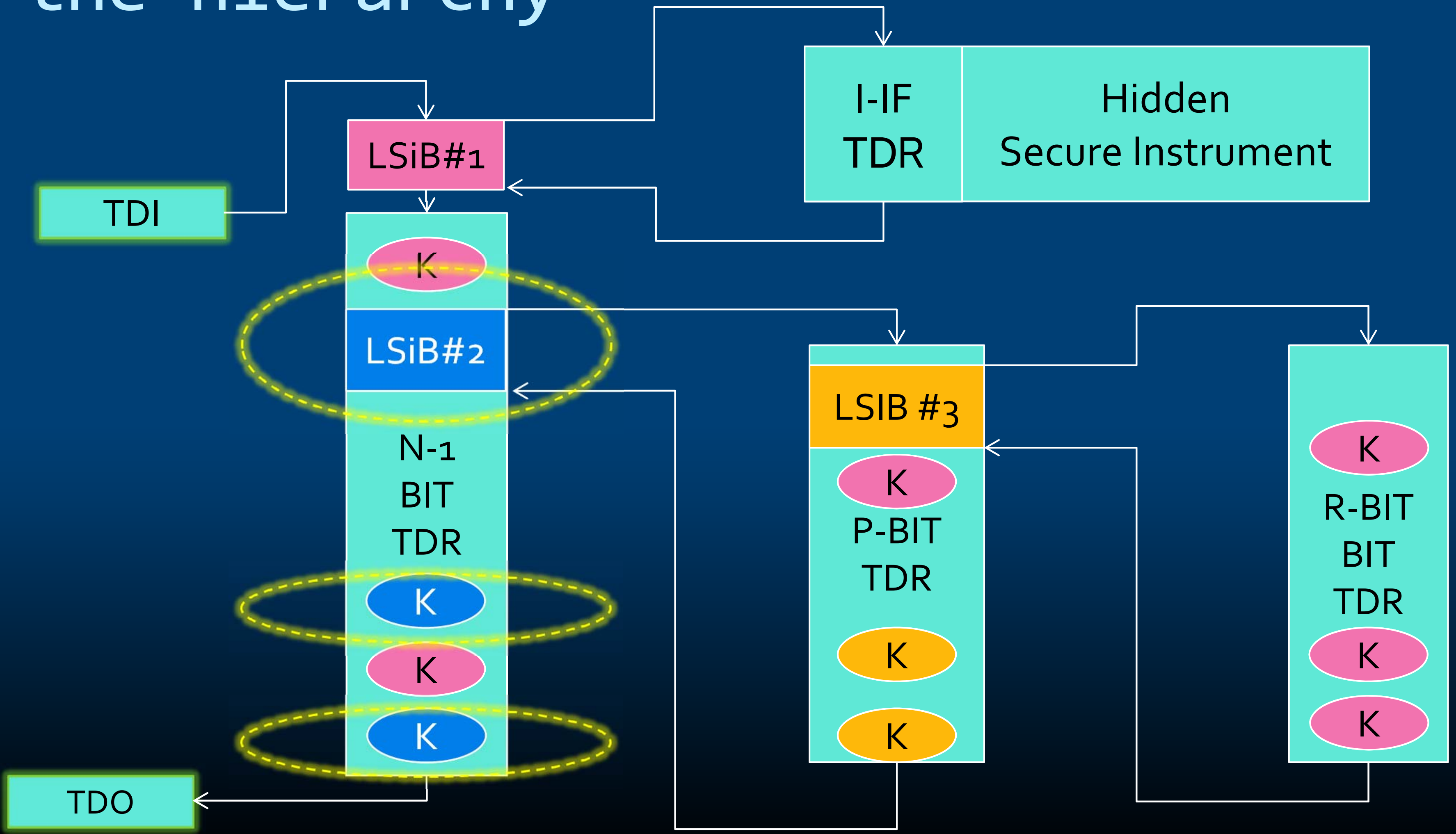
But…couldn’t they still get lucky and guess right the first time?

Yes…but what if they have to be lucky and guess right multiple times?

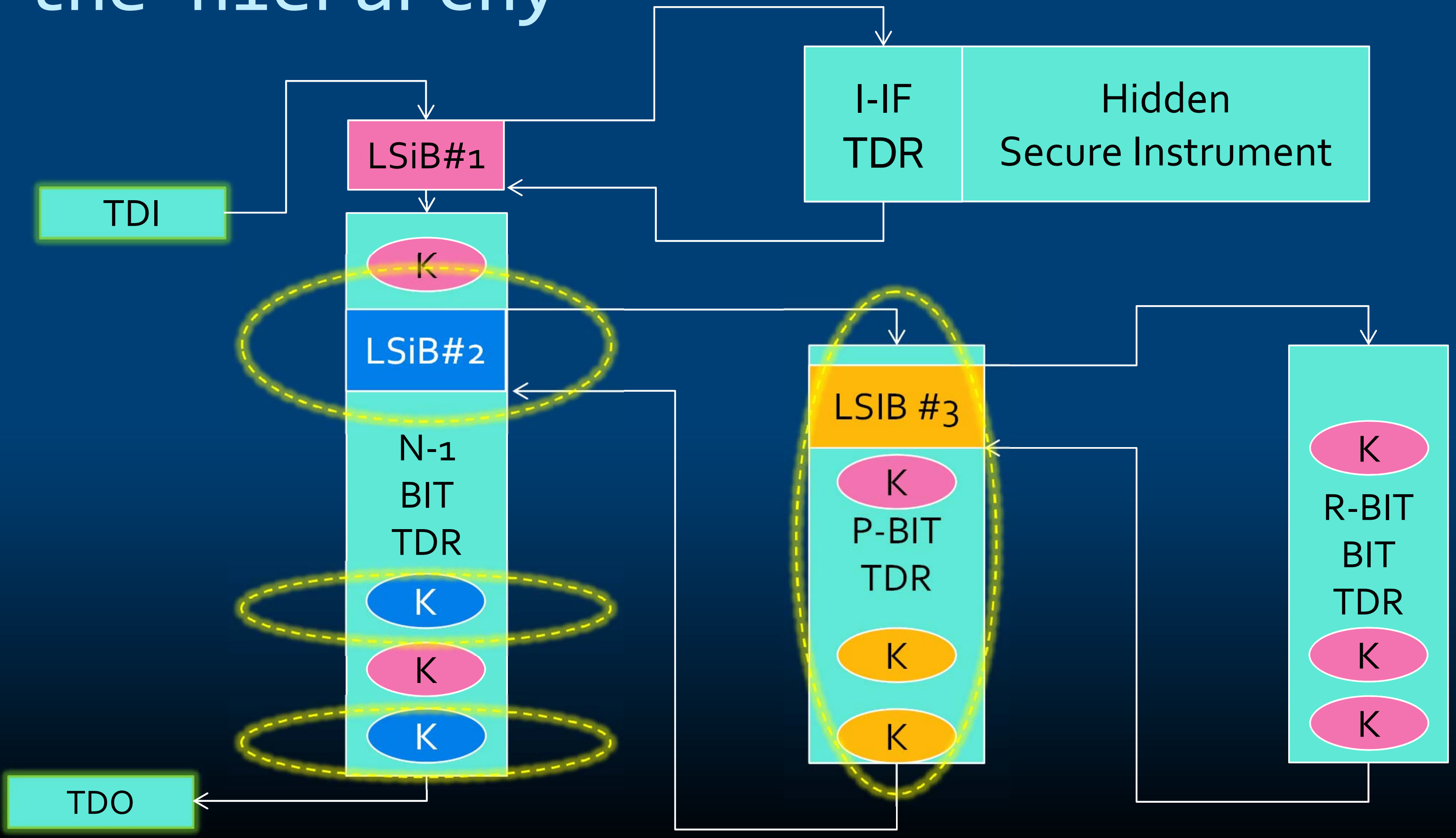
Multiple LSIBs, Keys, and Traps may be present on different levels of the hierarchy



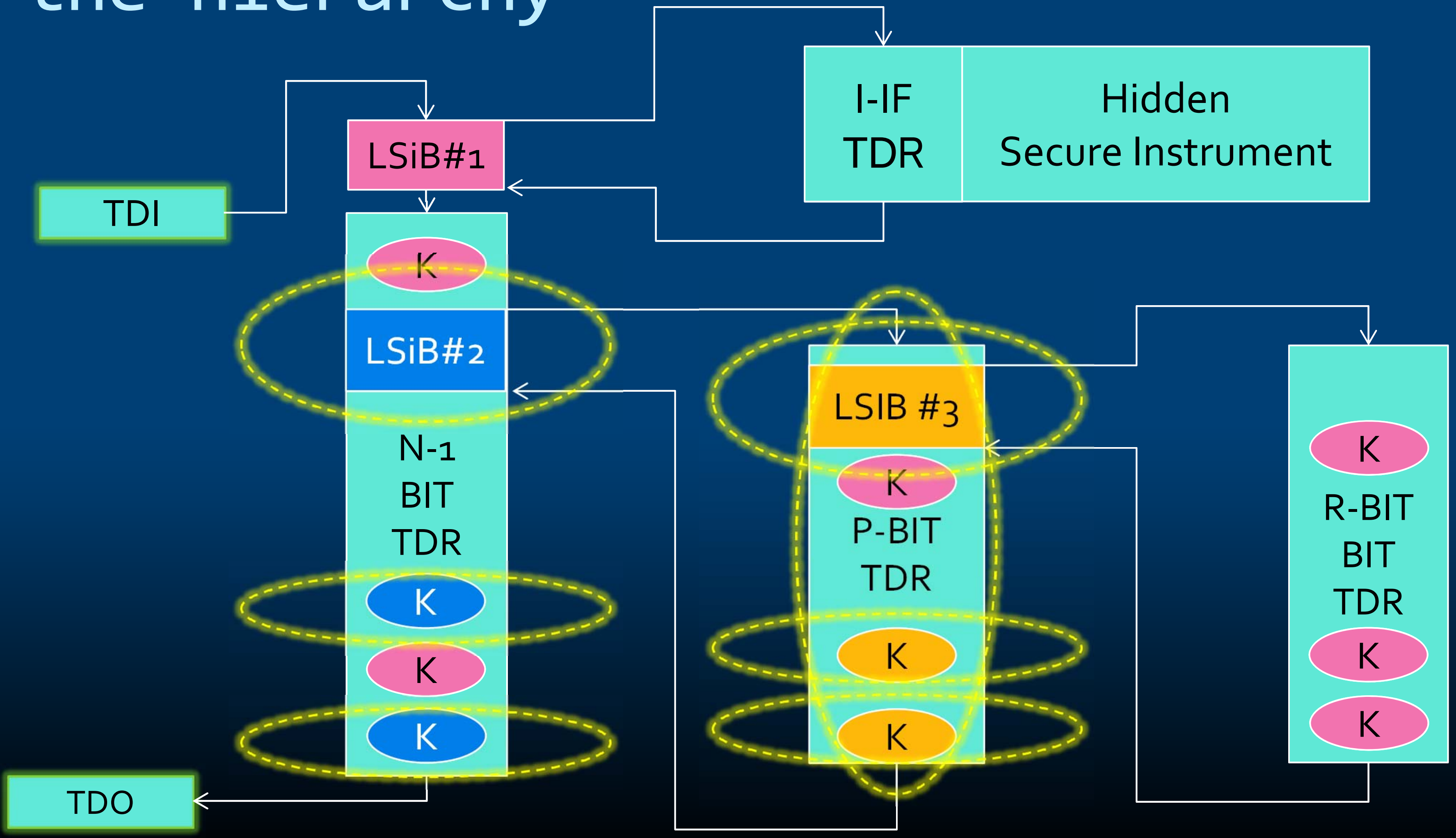
Multiple LSIBs, Keys, and Traps may be present on different levels of the hierarchy



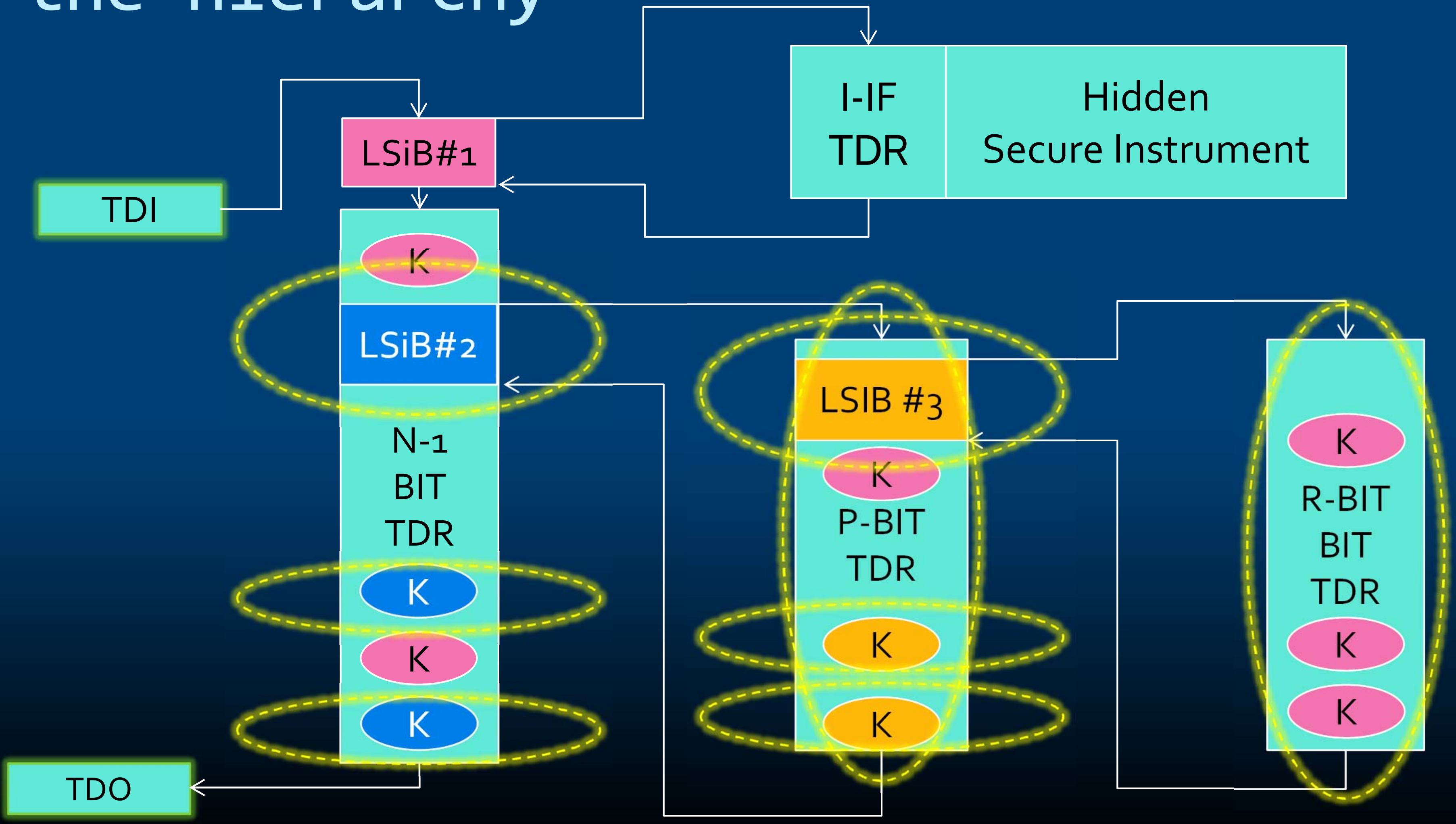
Multiple LSIBs, Keys, and Traps may be present on different levels of the hierarchy



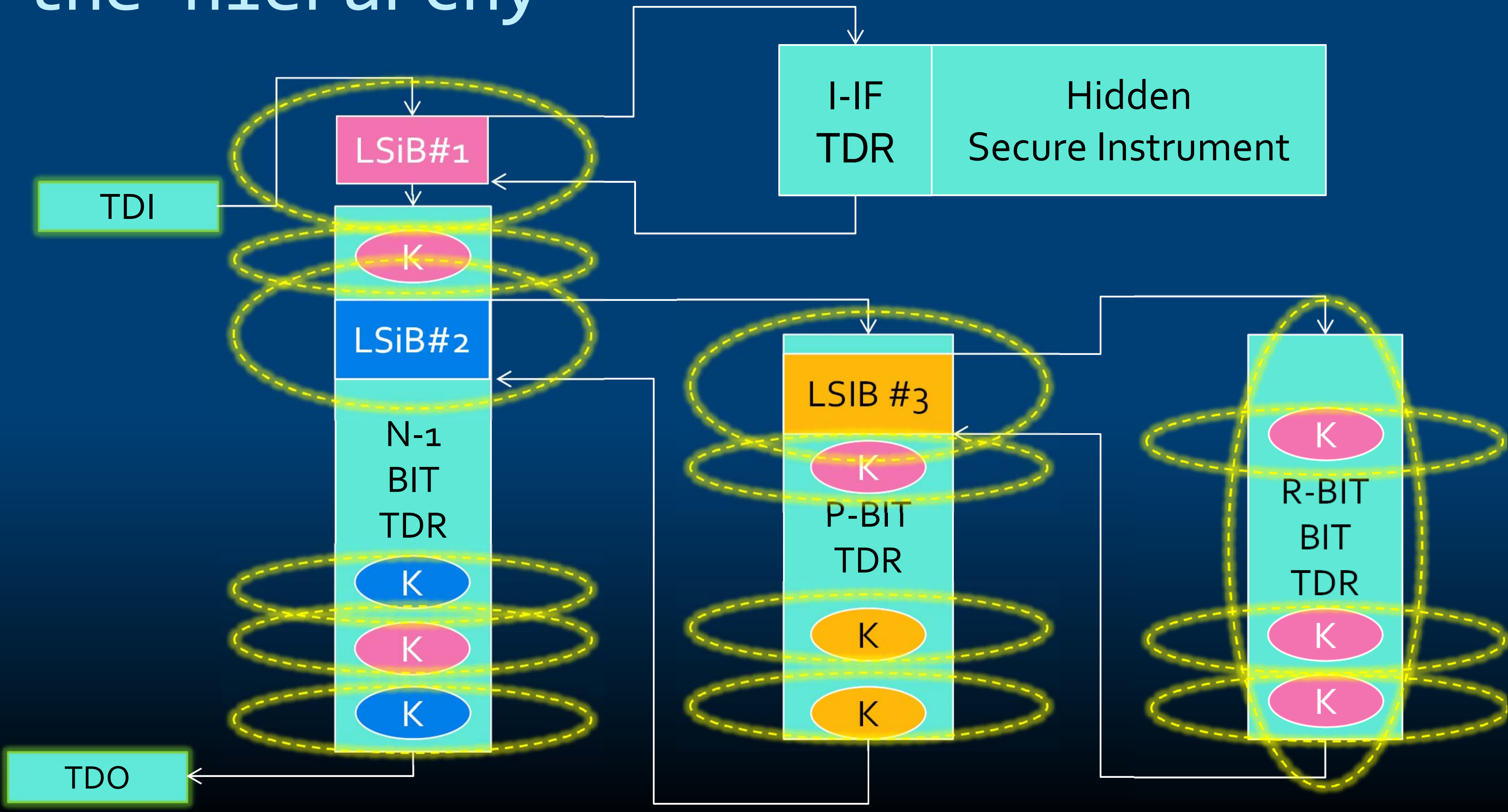
Multiple LSIBs, Keys, and Traps may be present on different levels of the hierarchy



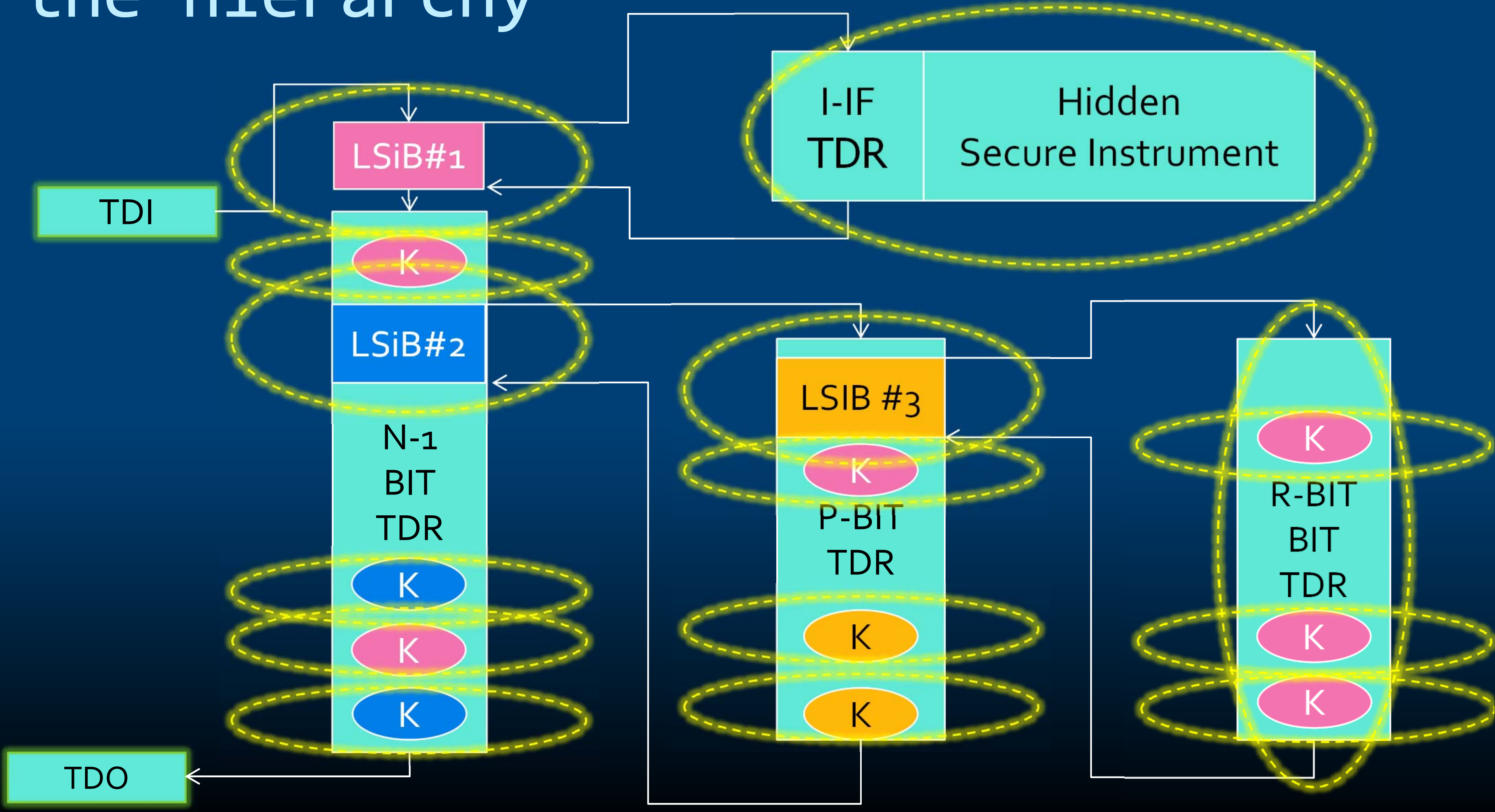
Multiple LSIBs, Keys, and Traps may be present on different levels of the hierarchy



Multiple LSIBs, Keys, and Traps may be present on different levels of the hierarchy



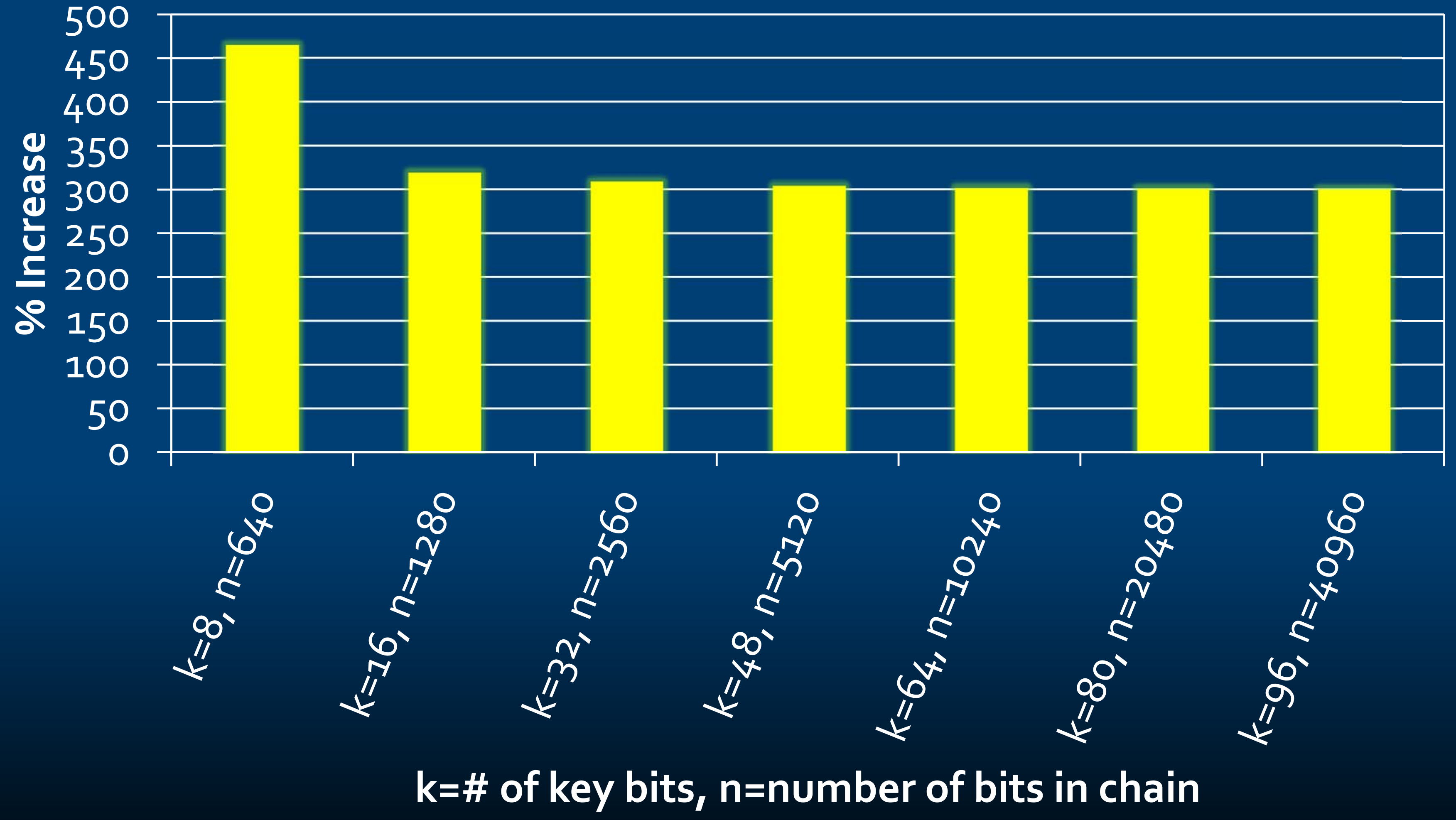
Multiple LSIBs, Keys, and Traps may be present on different levels of the hierarchy



Hierarchical Structure Increases Time for Access

- Multiple LSiBs may need to be unlocked in a particular order to access key bits.
- Trap Bits may be present at any level of hierarchy.
- Trap Bits can affect a different TDR than the one they reside in.
- Unlocking multiple LSiBs requires determining which bits are the actual LSIBs and keys at each step.

% Increase in Expected Time for 2 LSIBs over 1 LSiB with Traps



Conclusions

- Simply breaking the JTAG port to protect embedded instruments can severely hinder debug & diagnosis.

- Locking SIBs hide instruments by using data naturally present in the P1687 network as keys.

- Even relatively small key sizes can greatly increase expected time required for attack.

- Traps significantly increase the cost of a guess.

- Hierarchical structures with multiple LSIBs help protect against lucky first guesses and increase complexity.

- Locks, Keys, and Traps are an especially important tool for preventing counterfeiting, IP theft, and the malicious (or inadvertent) destruction of hardware.

Our work is supported in part by NSF grants CCF-1110290 and CCF-1061164 .