Adam Zygmontowicz, Micah Thornton, & Jennifer Dworak

Al Crouch & John Potter

SECURITY FOR CHIPS

Presented to SWDFT, June 7, 2013

Overview: Security needs to be considered in design and test of boards and ICs

- Trojans
- Counterfeits
- Side Channel Attacks
- IP and Data Protection

Waiting until you get attacked is too late!

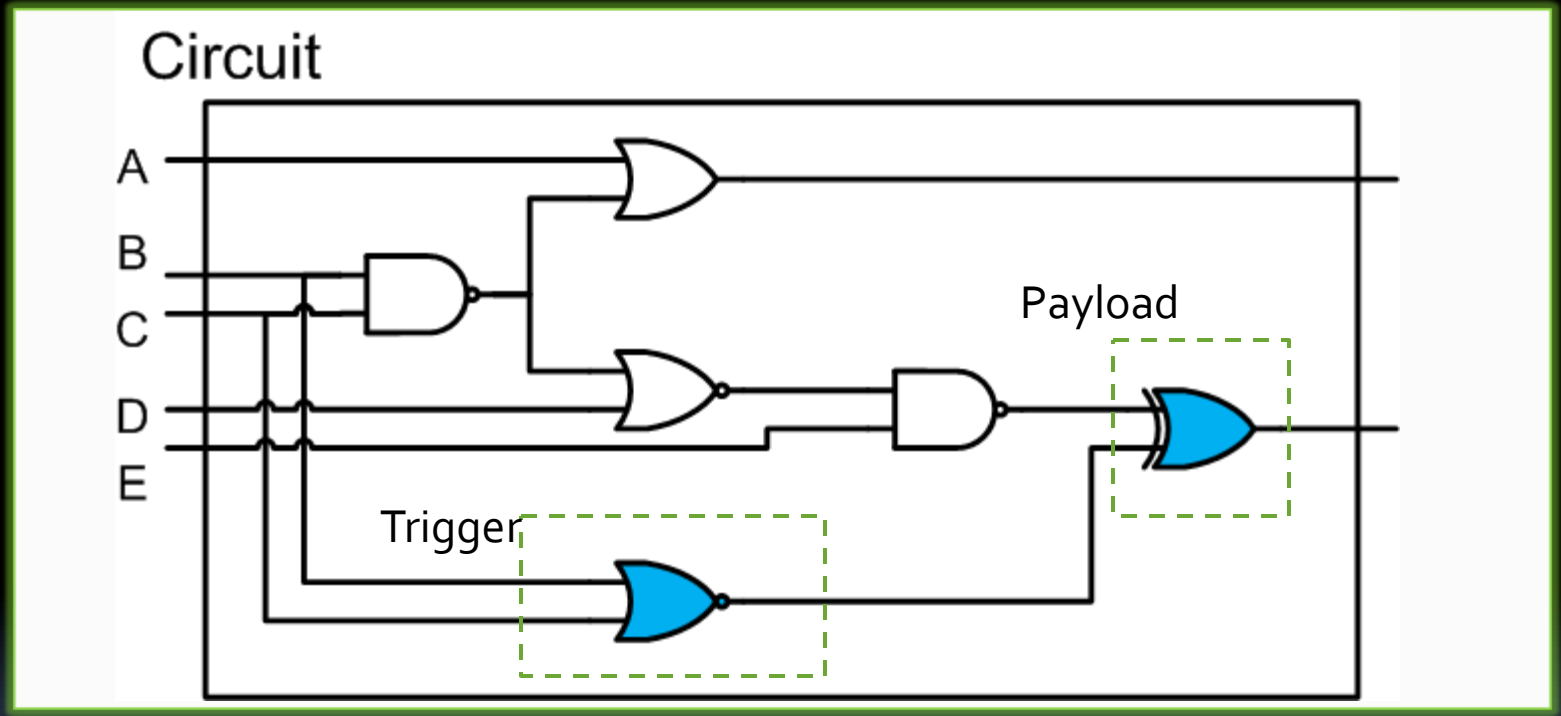
Overview: Security needs to be considered in design and test of boards and ICs

- Trojans
- Counterfeits
- Side Channel Attacks
- IP and Data Protection

Hardware Trojans

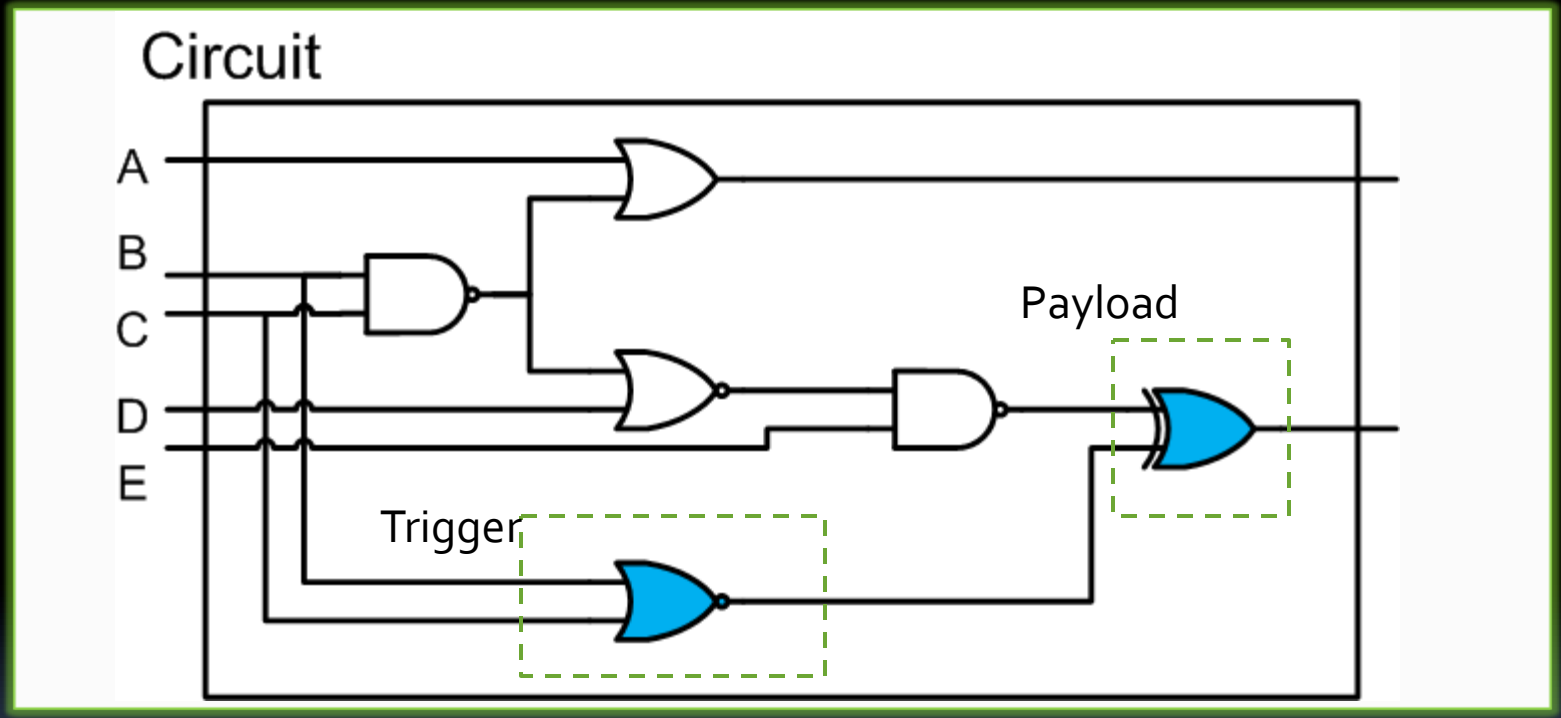
- Malicious changes to a design intentionally inserted by an attacker
- May be inserted at any stage of the design and manufacturing process: specification, RTL, manufacturing, supply chain
  - Most attention has focused on manufacturing
  - Design is important, too.
- Inserted with the intention of being stealthy
- Two components:
  - Trigger
  - Payload

Circuit with Combinational Trojan



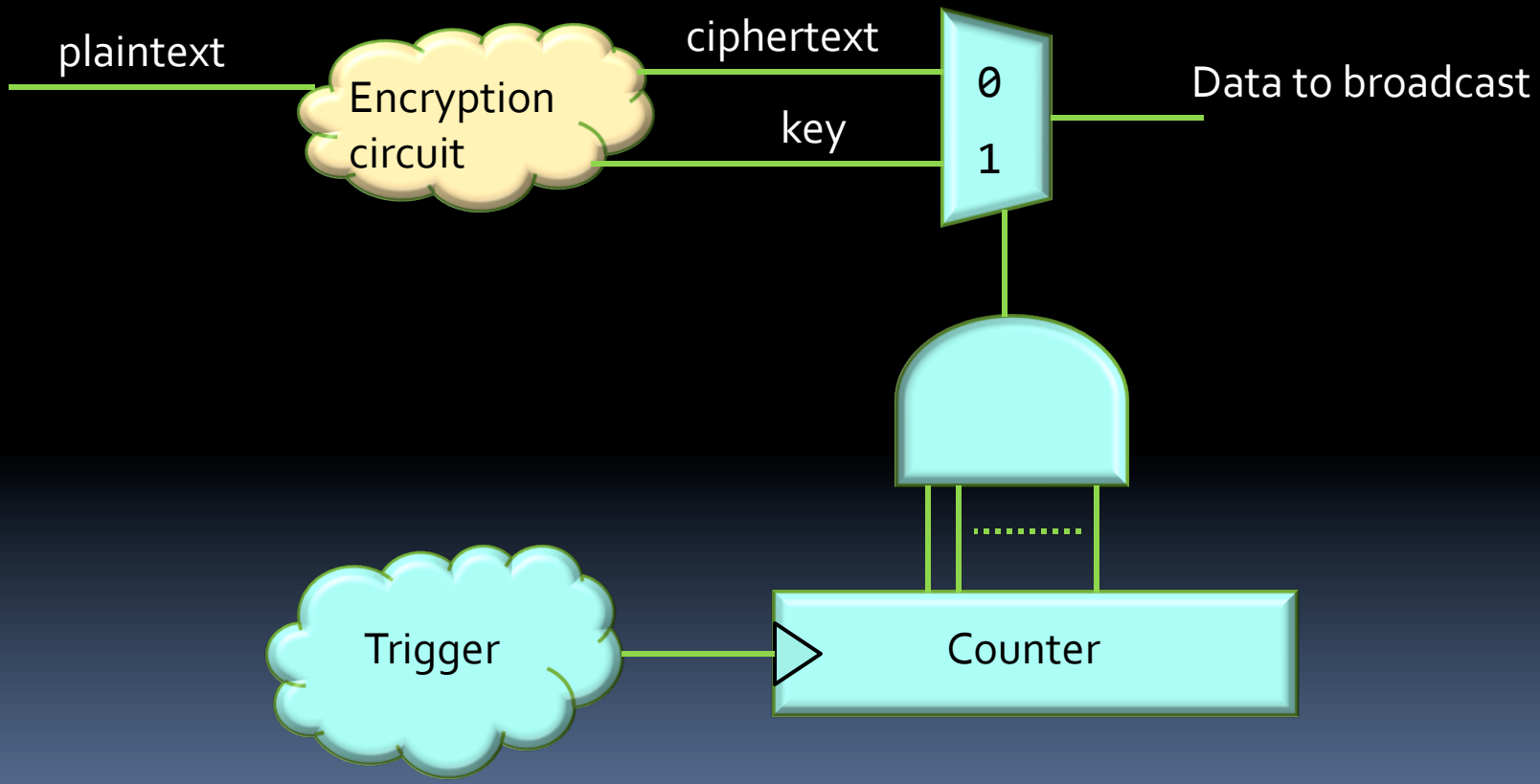
- Payload should affect something of functional importance to attacker
  - Leak Data
  - Cause Errors
  - Reduce Performance
  - Destroy the chip

Circuit with Combinational Trojan



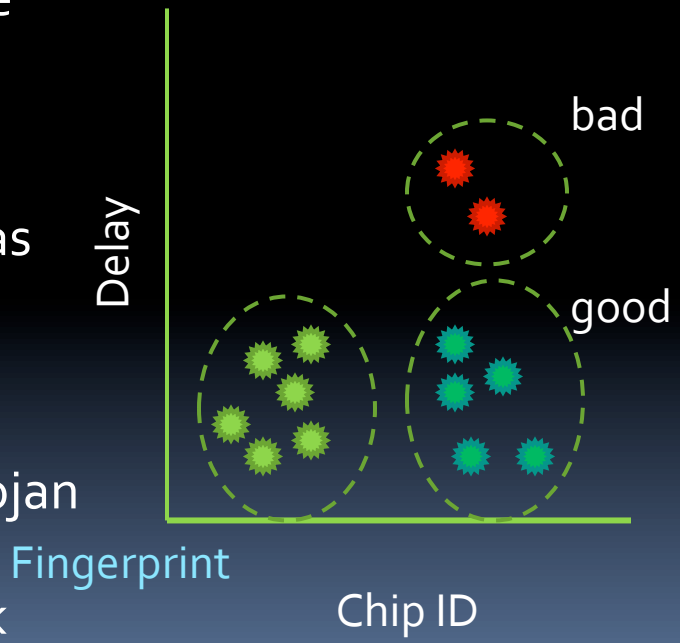
- Trigger should be stealthy
  - B=0, C=0 should be rare during functional operation
  - B=0, C=0 should not be targeted during structural test.

Sequential Trojan



How have researchers suggested that we detect Trojans inserted at manufacturing?

- Logic testing is generally ineffective
  - *Too hard to activate*
- Side channels affected by even inactive Trojans
  - Delay
  - Power
- Obtain "fingerprints" of chips verified as Trojan-free
- Process variations make comparison difficult
- Difference between Trojan and non-Trojan containing circuits is very small.
- Only works if Trojan is inserted at mask



Real Life Trojans….

before after

❖ On September 6, 2007, the Israeli Air Force carried out an airstrike on a Syrian nuclear reactor in Operation Orchard.

Hidden back door in microprocessors used in radar may have allowed them to be disabled remotely.

❖French microprocessors used in military applications have remote "kill switches" to allow them to be disabled.

❖During the Cold War, secret cameras were inserted inside Xerox 914 copy machines in the Soviet embassy to record copied documents.

Making *Good* Hardware Trojans is Non-trivial!

- This is especially true at manufacturing….
  - Hardware Trojans should have a functionally important effect.
  - The design needs to be reverse engineered *from the layout* to design and implement the Trojan.
- Insertion at the RTL (especially 3rd party IP) may be more troublesome….
  - Design reuse magnifies the impact
  - Easier to produce a functionally important effect
  - Need to get past design verification
  - We need to develop good ways to use test and verification to find these Trojans.

Making *Good* Hardware Trojans is Non-trivial!

- This is especially true at manufacturing….

Another place Trojans may appear: Counterfeit Circuits

- Design reuse magnifies the impact
- Easier to produce a functionally important effect
- Need to get past design verification
- We need to develop good ways to use test and verification to find these Trojans.

Making *Good* Hardware Trojans is Non-trivial!

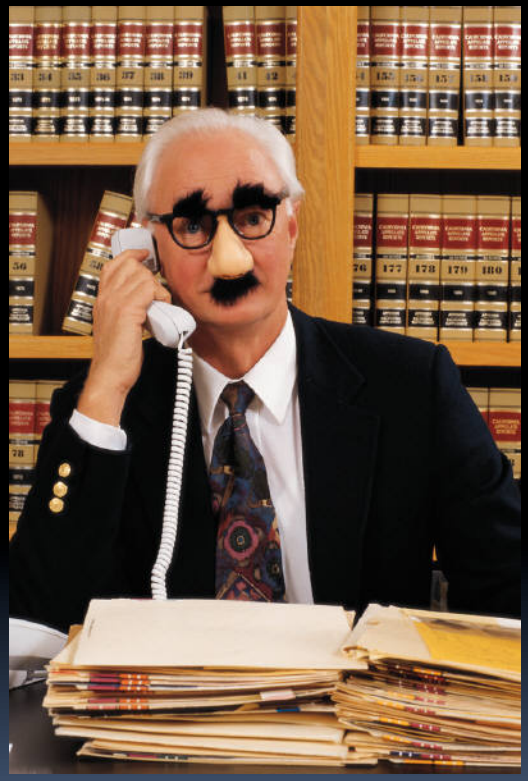
- This is especially true at manufacturing….
  - Hardware Trojans should have a functionally important effect.
  - The design needs to be reverse engineered *from the layout* to design and implement the Trojan.
- Insertion at the RTL (especially 3rd party IP) may be more troublesome….
  - Design reuse magnifies the impact
  - Easier to produce a functionally important effect
  - Need to get past design verification
  - We need to develop good ways to use test and verification to find these Trojans.

Overview: Security needs to be considered in design and test of boards and ICs

- Trojans
- Counterfeits
- Side Channel Attacks
- IP and Data Protection

Types of Counterfeits

- Reverse engineer, design, and manufacture chips to be functionally similar to the original
- Salvage old chips from boards and sell them as new chips
- Re-label low-performing die as high-performing
- Sell defective parts as working chips



Why should industry care about counterfeits?

- Less reliable than valid die
- Harms the reputation of the real chip/board provider
- Denies revenue to original chip/board provider
- Increases support costs – the counterfeit die and boards may require support or may be returned
- May contain malicious functionality
- DoD Proposed Rule: "Defense Federal Acquisition Regulation Supplement: Detection and Avoidance of Counterfeit Electronic Parts"
  - Aims to shift liability for replacing counterfeit parts to contractors

Selected Counterfeit Incidents

- Between 2007 and 2010 over **5.6 million** counterfeit semiconductor devices were seized by Customs and Border Patrol (CBP) and ICE (Immigrations and Custom Enforcement)
- **IHS iSuppli** estimates that the total reported incidents for counterfeit parts is approximately **12 million** over the last 5 to 6 years. This averages to one counterfeit part found every 15 seconds.
- In 2009, a NASA probe project was delayed nine months and went 20% over-budget due partly to counterfeit parts.
- Entire NEC product line was counterfeited in across multiple factories in China and Taiwan
- Display unit manufactured by L-3 Communications that provides the pilot with information regarding fuel use, engine status, and warnings was found to contain a chip that had a 27% increase in failure rates. Over 400 assemblies were affected.
- VisionTech imported more than 3200 identified or suspected shipments of counterfeit microelectronics to the U.S.
  - Sold to military for use in missile targeting systems, identification friend-or-foe systems, among others
  - Thousands of parts may still be in the supply chain

VisionTech's Cost to Companies

AMD	$34.9K	National Semiconductor	$5.9K
Altera	$7.6K	NEC	$24.8K
Analog Devices	$75.6K	Peregrine Semiconductor	$2.6K
Cypress Semiconductor	$33.4K	Phillips Electronics	$1.6K
Freescale	$40K	Renesas	$2.4K
Infineon Technologies	$10K	Samsung Elect. America	$77.2K
Intel	$100.9K	STMicroelectronics	$18.6K
Intersil	$1.9K	Texas Instruments	$92.9K
Linear Technology	$32K	Toshiba	$2.4K
Maxim	$1.6K	Xilinx	$22.2K
Mitel	$2.6K	**Total**	**$591.4K**

Detecting & Avoiding Counterfeits

- F
  i
  c
- E
- I
- S
- I
- I
  t
- F
  u

t.

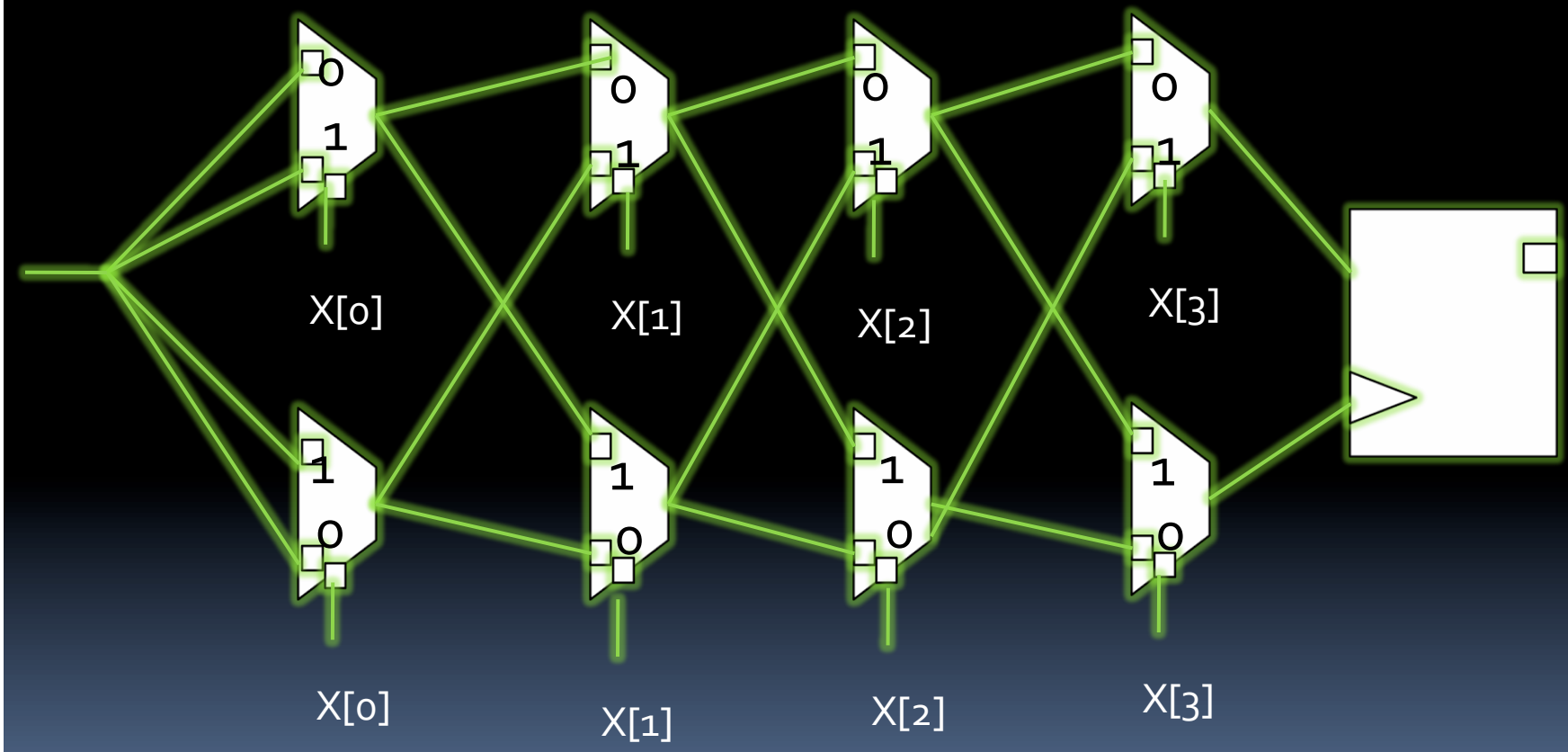
Detecting & Avoiding Counterfeits

- Reporting suspected or discovered counterfeit incidents to an anti-counterfeiting clearinghouse
- Buying from authorized suppliers
- Inspection of packaging
- Sensors to detect aging
- Incoming test
- Device authentication (e.g. with die IDs and a trusted database.)
- Hiding information in chips to make unauthorized access and copying more difficult.

Cloning can defeat the ID approach

- If ID is programmed into the device, an attacker can replicate that ID in their own copy.
- Physically Unclonable Functions (PUFs) try to address this:
  - Visual IDs (Randomly Scattered Phosphor Particles)
  - Electronic PUFs (take advantage of process variations: ring oscillator PUFs, SRAM PUFs, Arbiter PUFs)

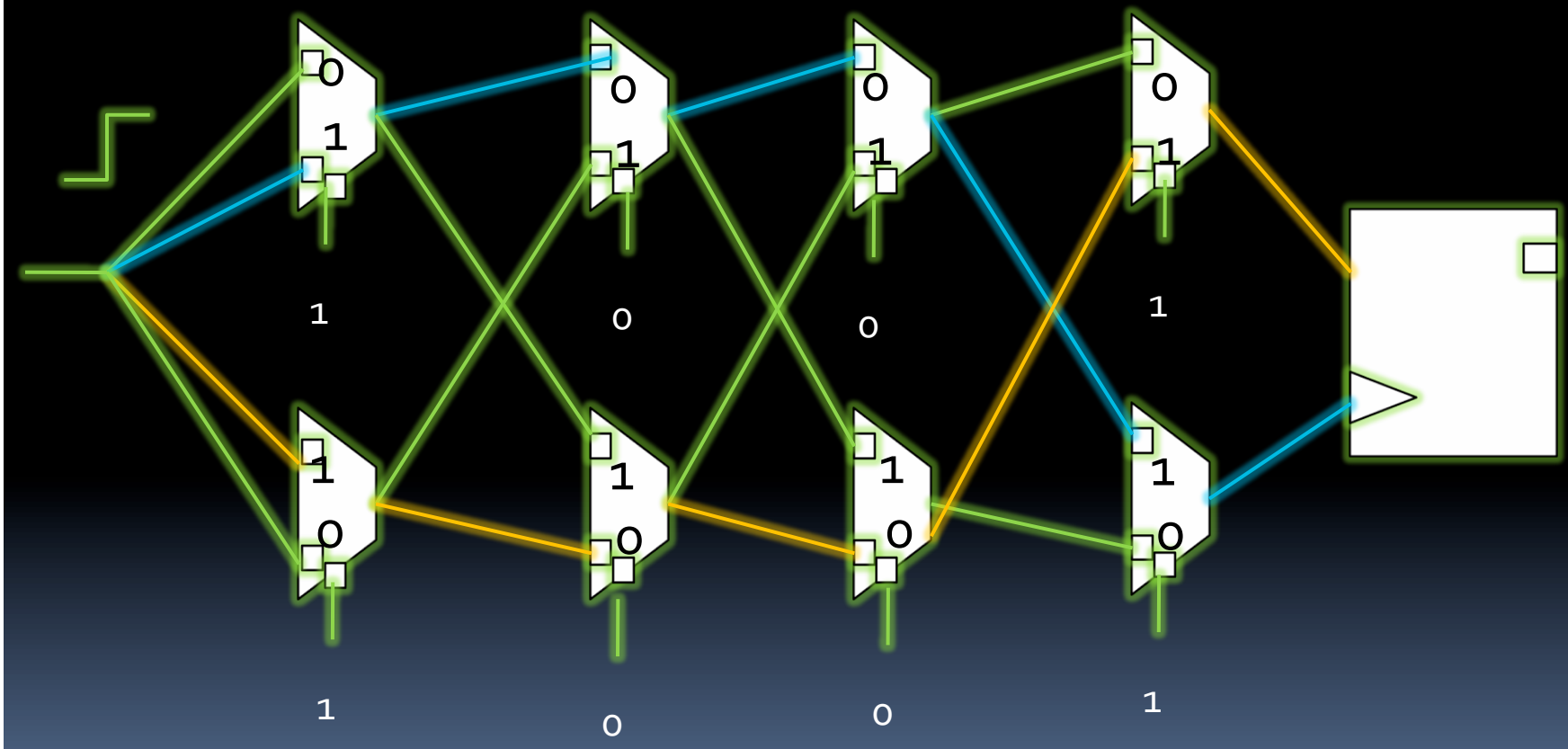
Example PUF: Arbiter PUF



Suh and Devadas, "Physical Unclonable Functions for Device Authentication and Secret Key Generation", 2007

Example PUF: Arbiter PUF

Challenge = 1001



If the orange path is faster, 1 is latched. If the blue path is faster, 0 is latched.

Some Up and Coming Tools & Standards for Avoiding Counterfeiting

- iNemi Tools
  - Risk of Counterfeit Use, Risk of Untrusted Sources and Counterfeit Loss & Total Cost Estimations
- SAE International: WIP standard for Fraudulent/Counterfeit Electronic Parts; Avoidance, Detection, Mitigation, and Disposition
  - Originally focused on aerospace industry and now expanded to all electronic part components

Overview: Security needs to be considered in design and test of boards and ICs

- Trojans
- Counterfeits
- Side Channel Attacks
- IP and Data Protection

Side Channel Attacks

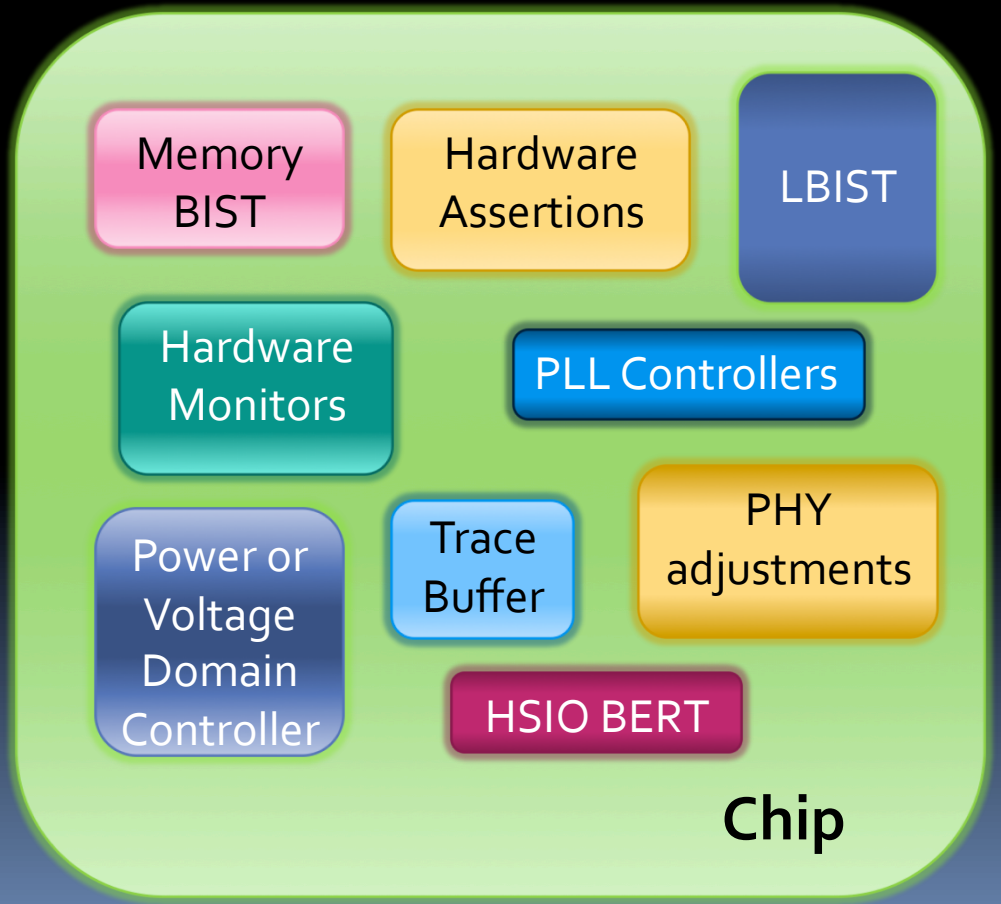
- Break encryption algorithms by using power or delay traces to guess the key
- Can be very powerful technique
- In 2011, researchers in Italy and Germany used the power trace of the bootup cycle to break bitstream encryption of Xilinx Virtex 2 FPGA's in a matter of hours.
  - Allows someone to steal design IP and replace good FPGA bitstream with a Trojan one.

Overview: Security needs to be considered in design and test of boards and ICs

- Trojans
- Counterfeits
- Side Channel Attacks
- IP and Data Protection

On-Chip Instruments need 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

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

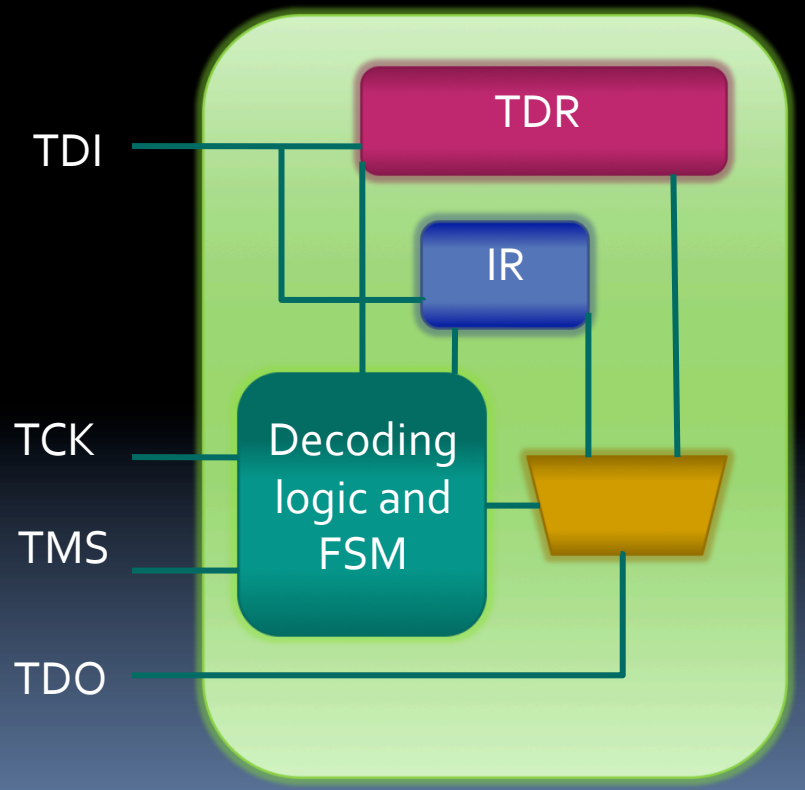
We need a way to prevent unauthorized users from accessing this information through the scan chain.

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.

Internals of chip/board can be accessed with IEEE 1149.1

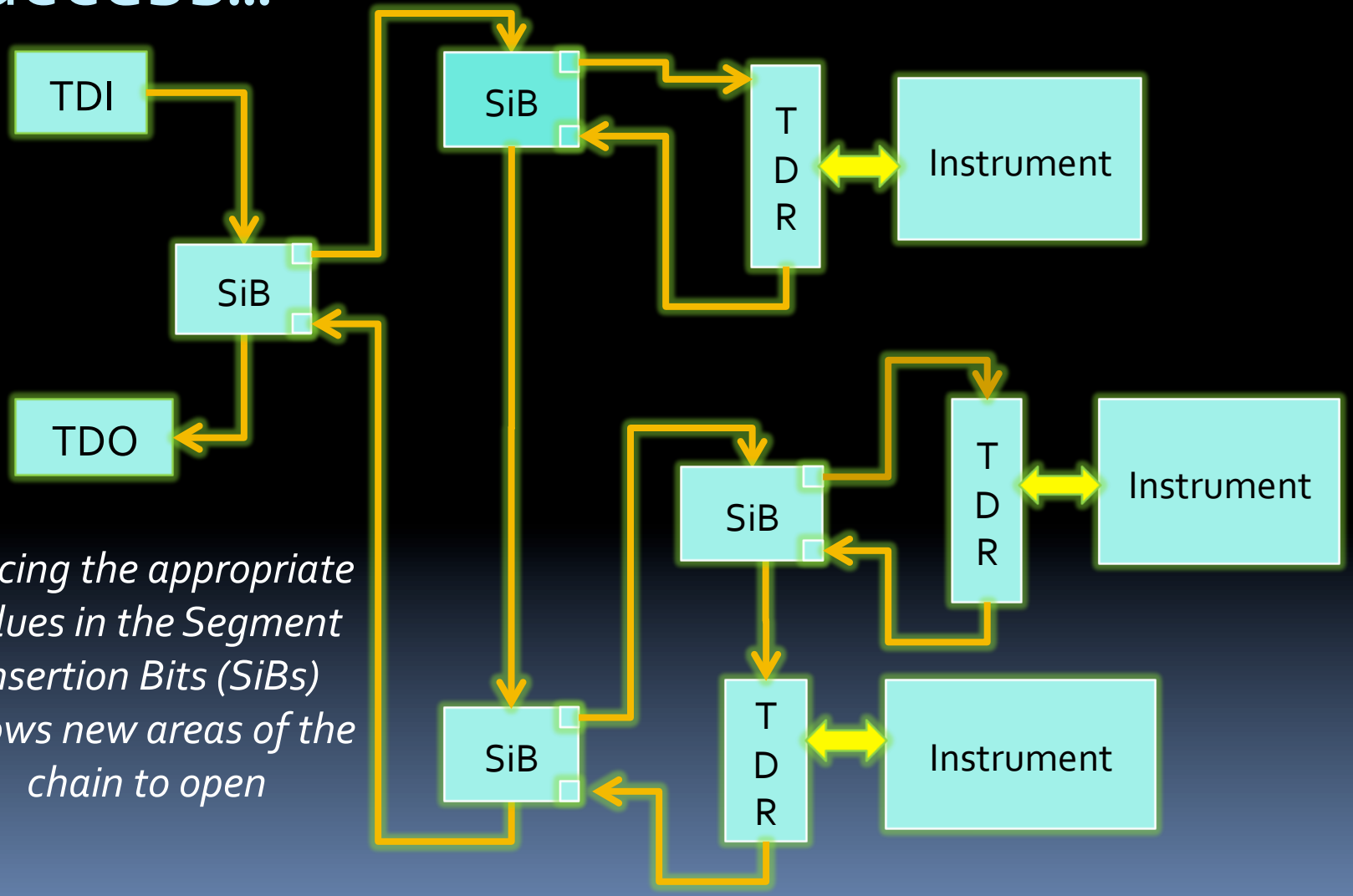
- Attackers can hack the JTAG port
  - Modify firmware (Xbox Hack)
  - Search for "private" instructions that are hidden test and debug features
  - Altera lists private instructions to avoid or you will destroy the chip!



How can information be protected in 1149.1?

- Some companies burn out the JTAG port by fusing off the TMS signal.
  - Problem: Can no longer use JTAG for debug or field test.
- Other researchers have suggested 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 asses privilege level.
    - UpdateDR is blocked if user doesn't have privileges

P1687 is designed for instrument access…

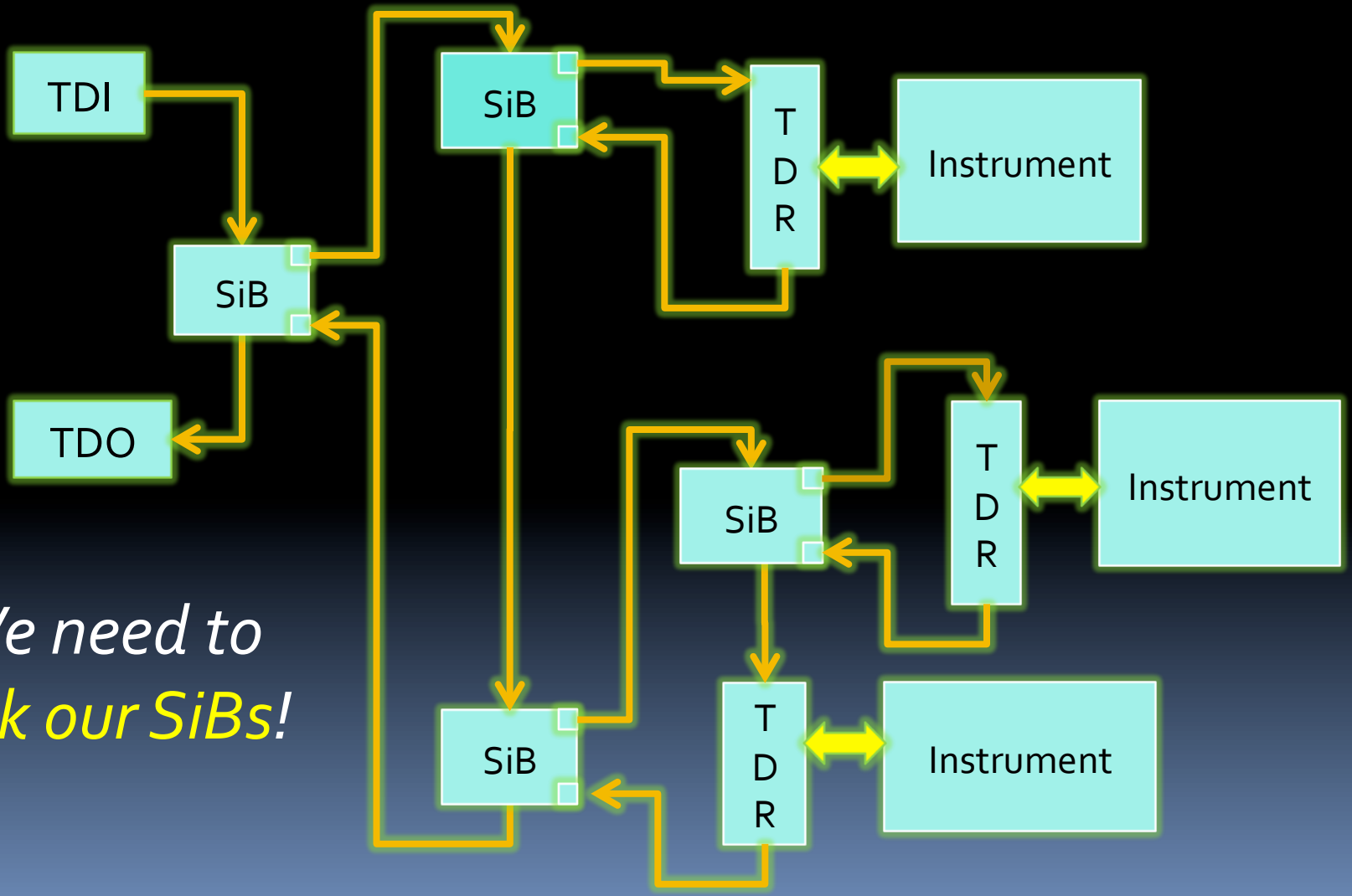


Placing the appropriate values in the Segment Insertion Bits (SiBs) allows new areas of the chain to open

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

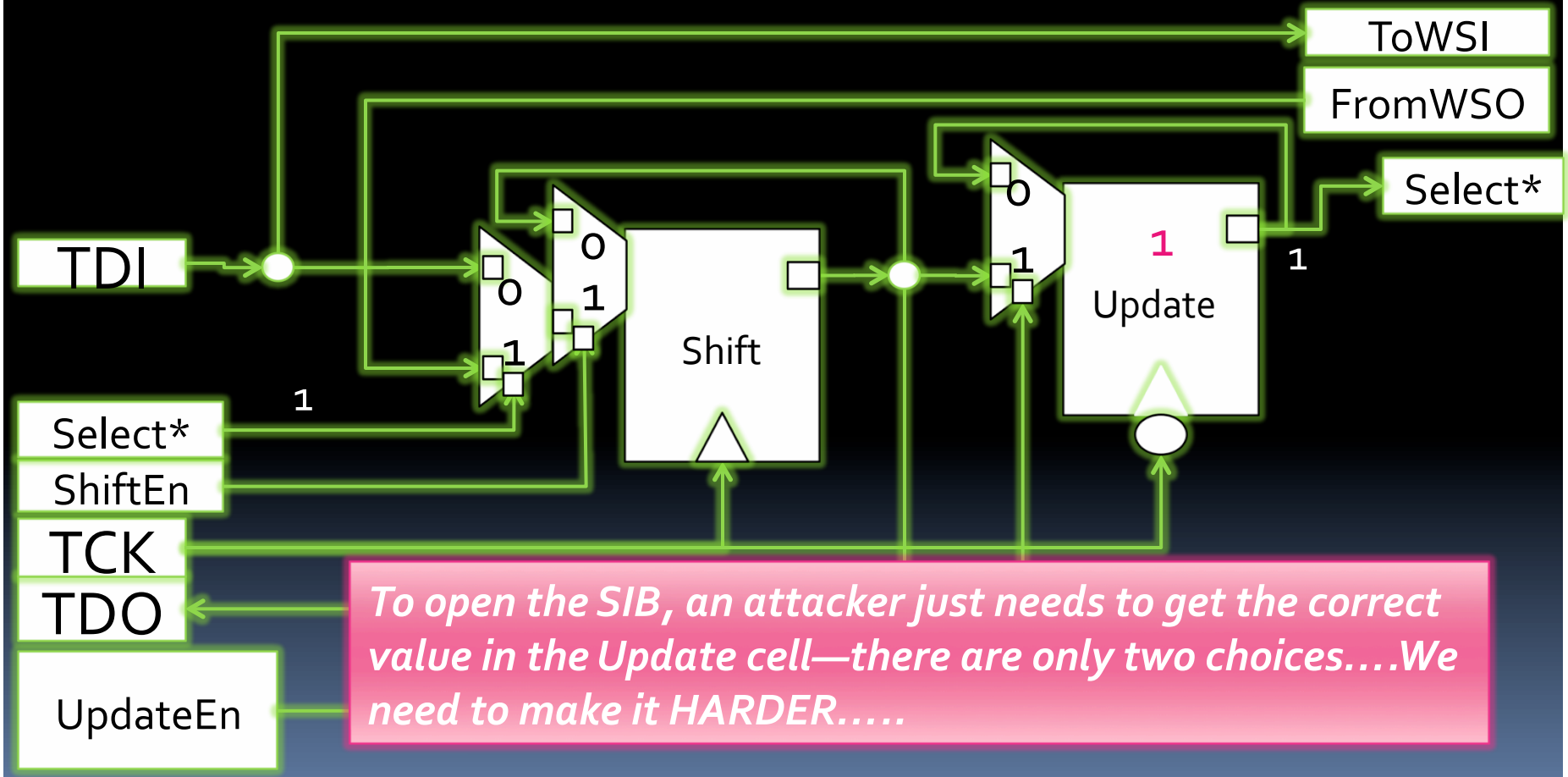
Yes!

SiBs serve as gates that must be opened for instrument access…



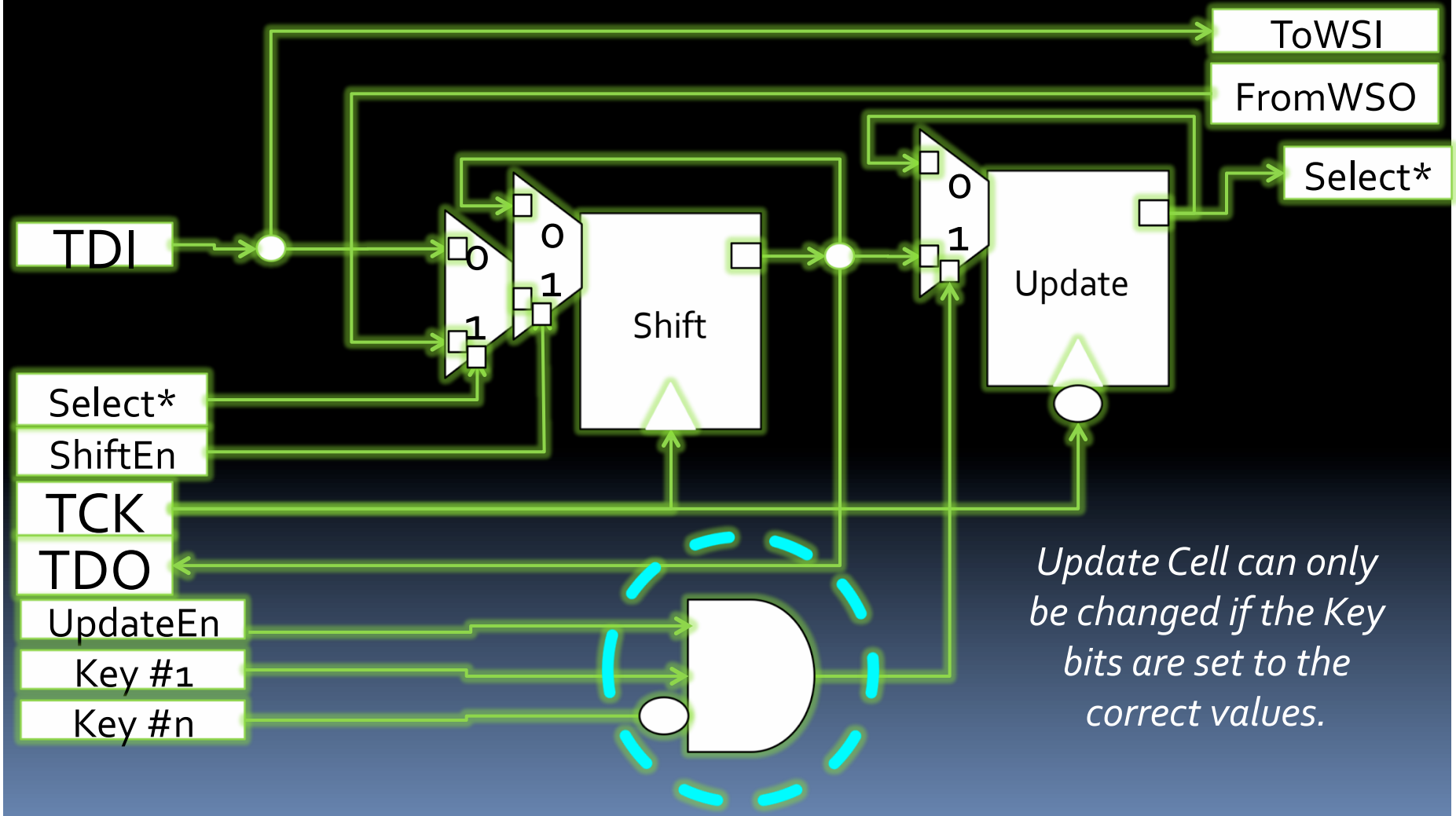
We need to lock our SiBs!

What does a normal SiB look like?

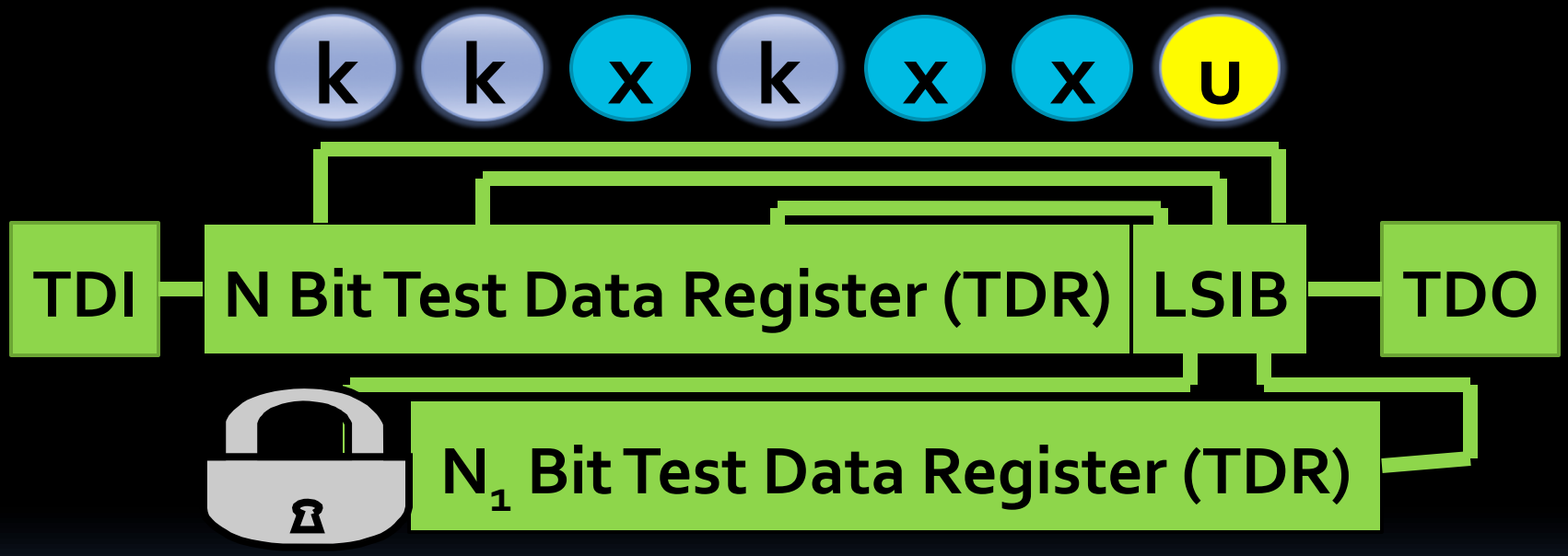


To open the SIB, an attacker just needs to get the correct value in the Update cell—there are only two choices….We need to make it HARDER…..

A Locking Segment Insertion Bit (LSiB)



The attacker must guess the right key bits to open the LSIB…

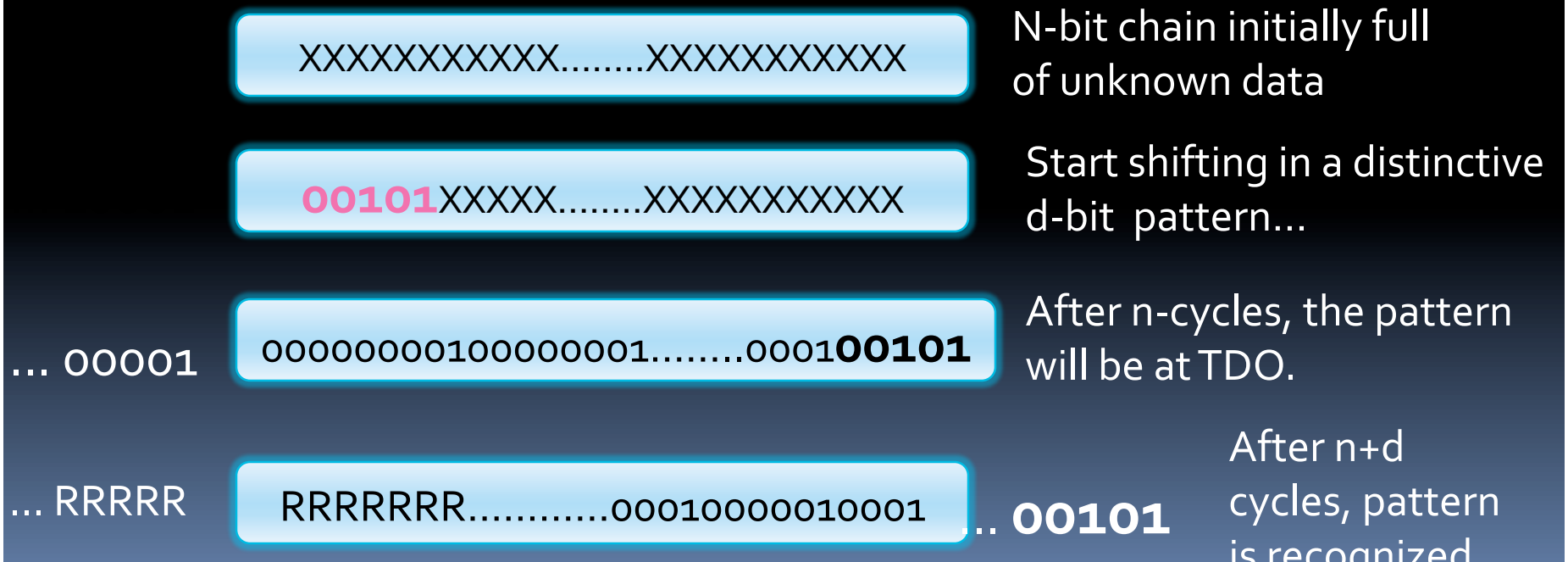


If the attacker doesn't know anything about the network, his best guess (to avoid bias) will be random….

How long is guessing likely to take?

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

How will the attacker know the length of the chain?



How does the attacker make a guess?

The attacker must use 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…

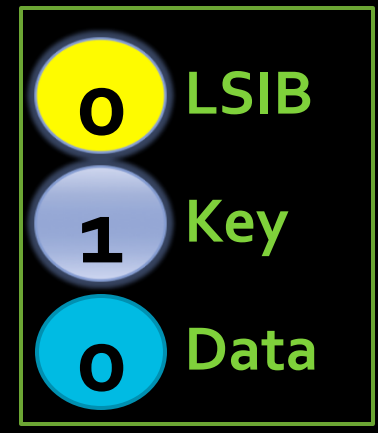
$5+n+d$

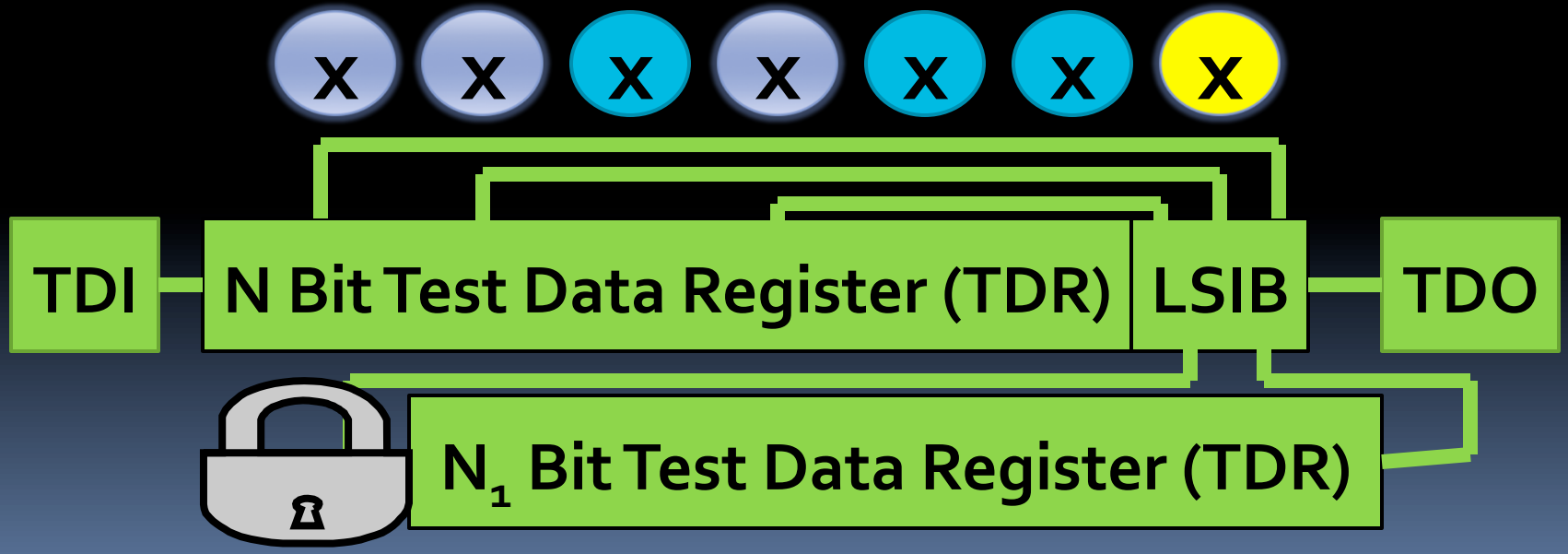
How Attackers Unlock LSIBs

Needed Key = 101

First guess = 1000010

Distinguishing seq = 0100101





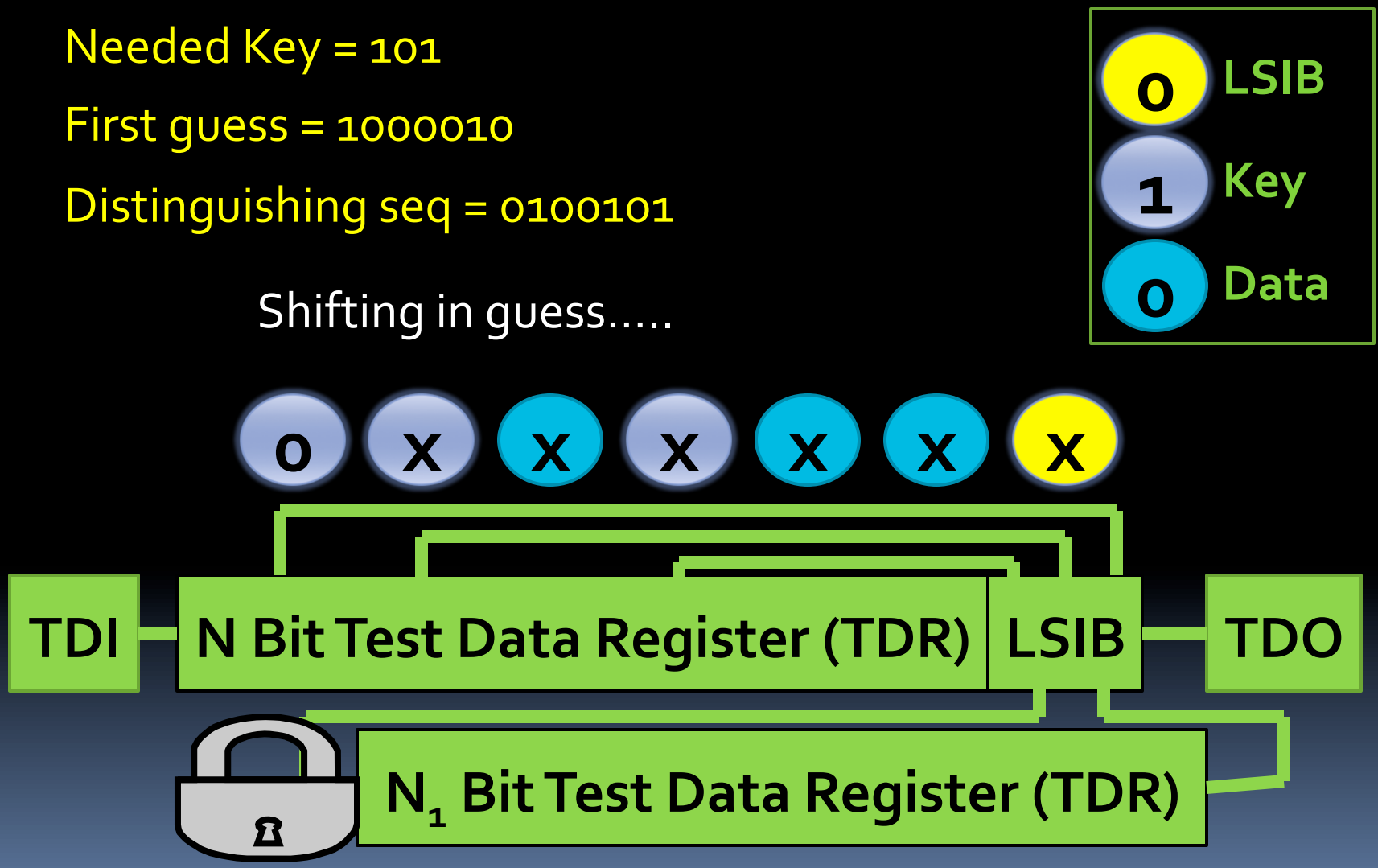
How Attackers Unlock LSIBs

Needed Key = 101

First guess = 1000010

Distinguishing seq = 0100101

Shifting in guess…..

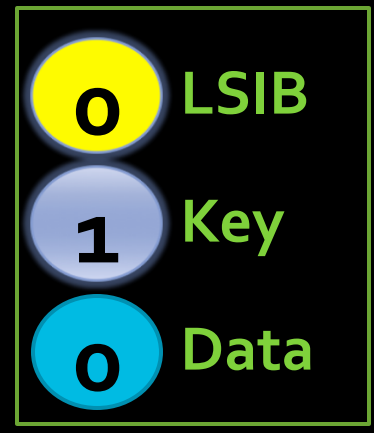


How Attackers Unlock LSIBs

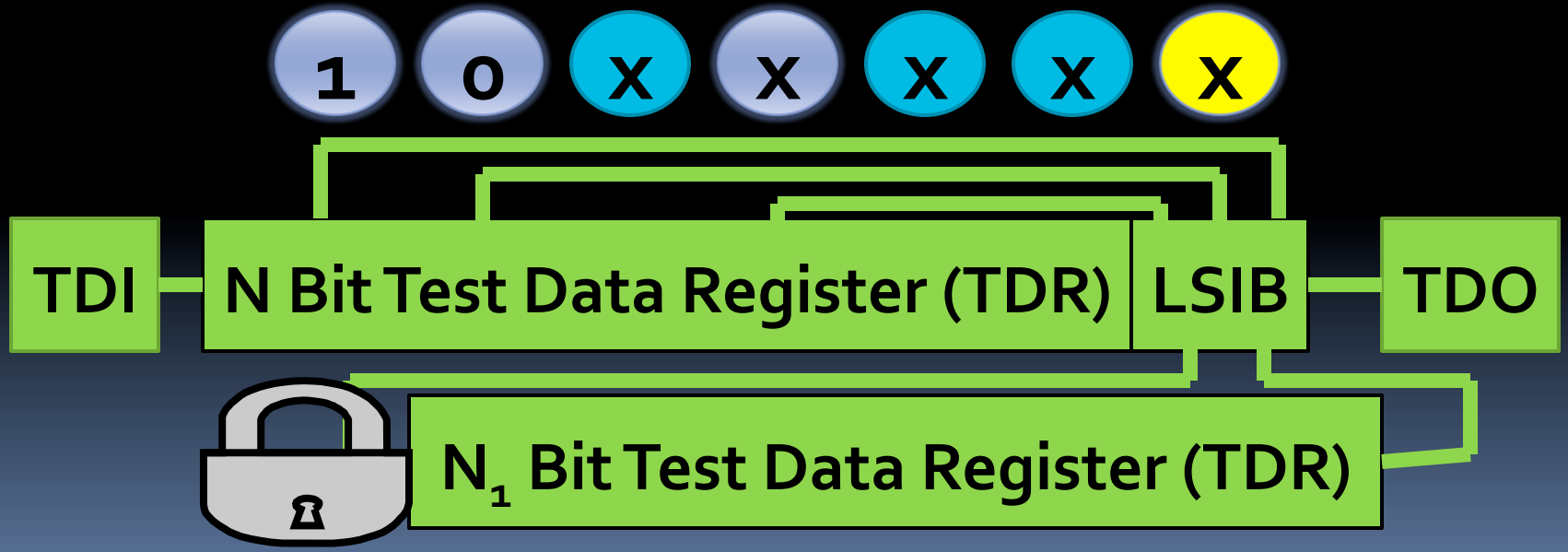
Needed Key = 101

First guess = 1000010

Distinguishing seq = 0100101



Shifting in guess…..



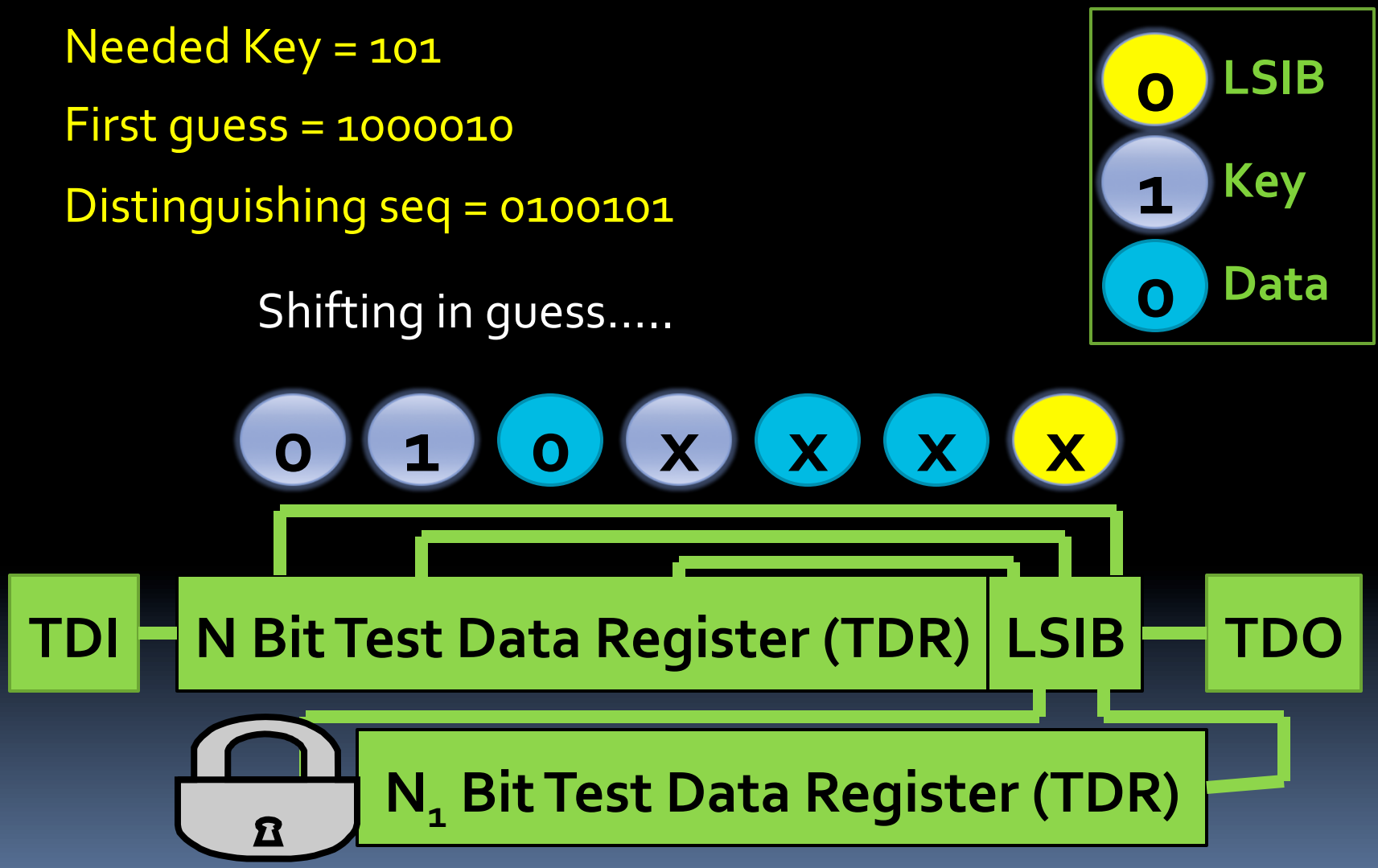
How Attackers Unlock LSIBs

Needed Key = 101

First guess = 1000010

Distinguishing seq = 0100101

Shifting in guess…..



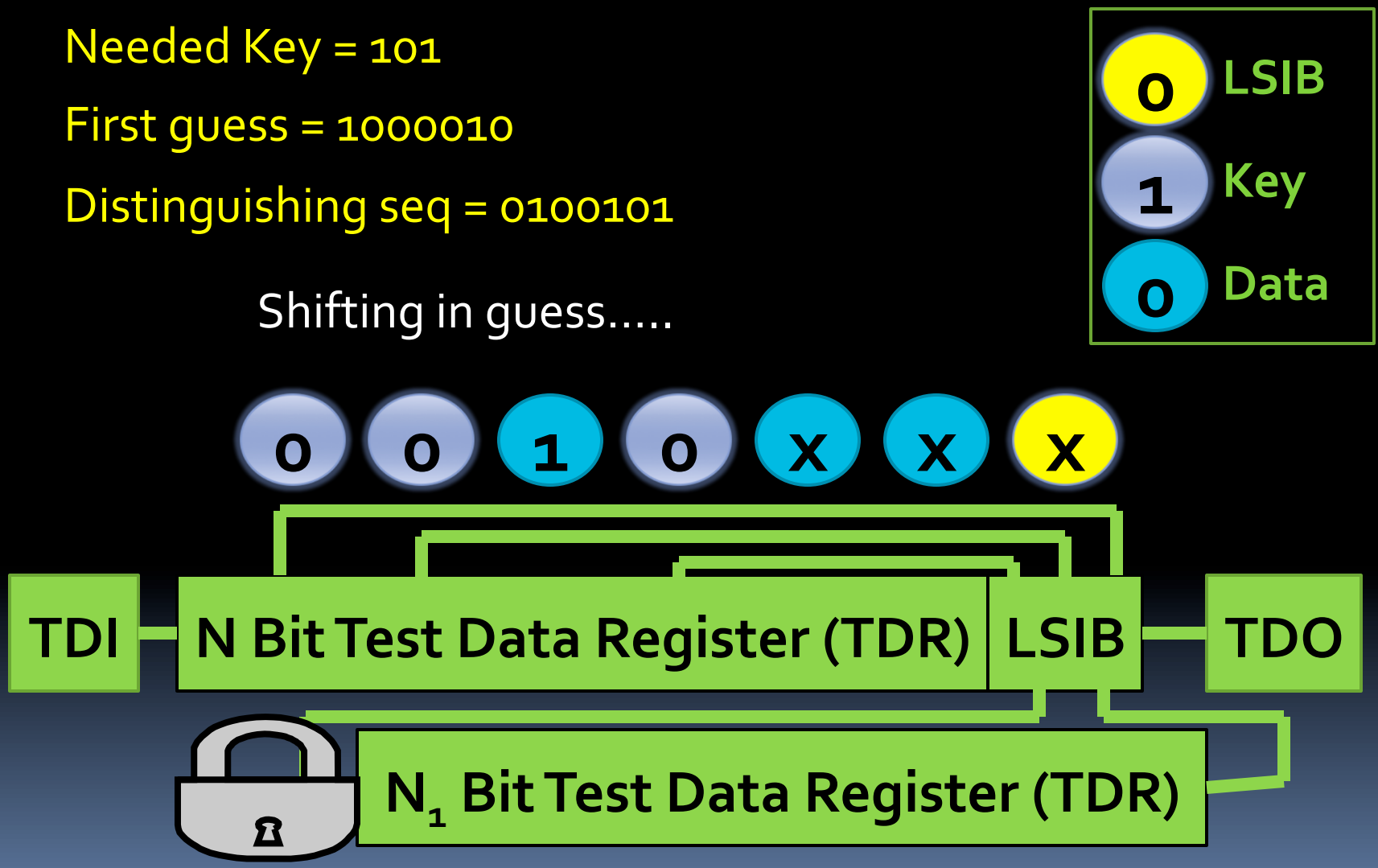
How Attackers Unlock LSIBs

Needed Key = 101

First guess = 1000010

Distinguishing seq = 0100101

Shifting in guess…..



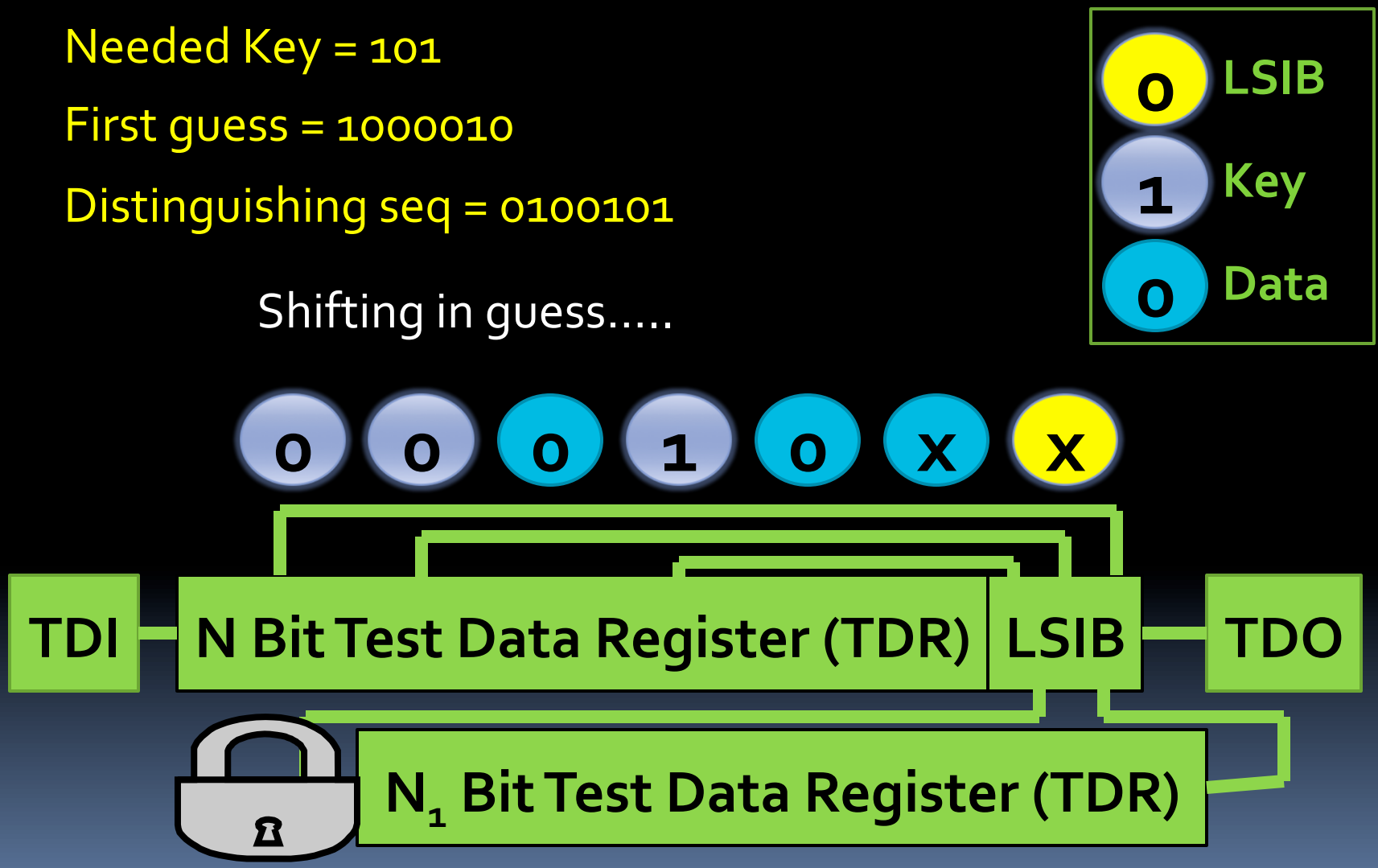
How Attackers Unlock LSIBs

Needed Key = 101

First guess = 1000010

Distinguishing seq = 0100101

Shifting in guess…..



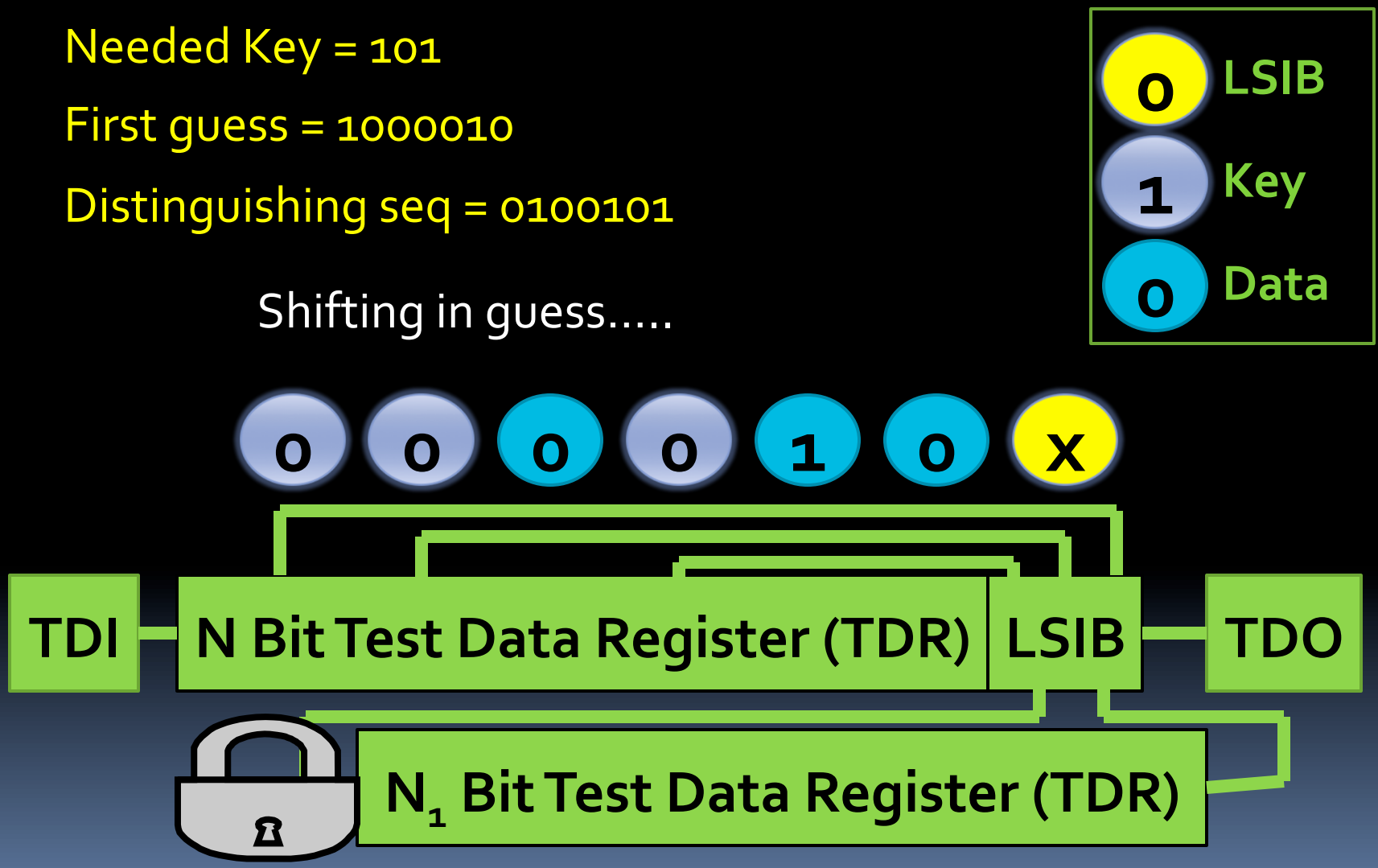
How Attackers Unlock LSIBs

Needed Key = 101

First guess = 1000010

Distinguishing seq = 0100101

Shifting in guess…..



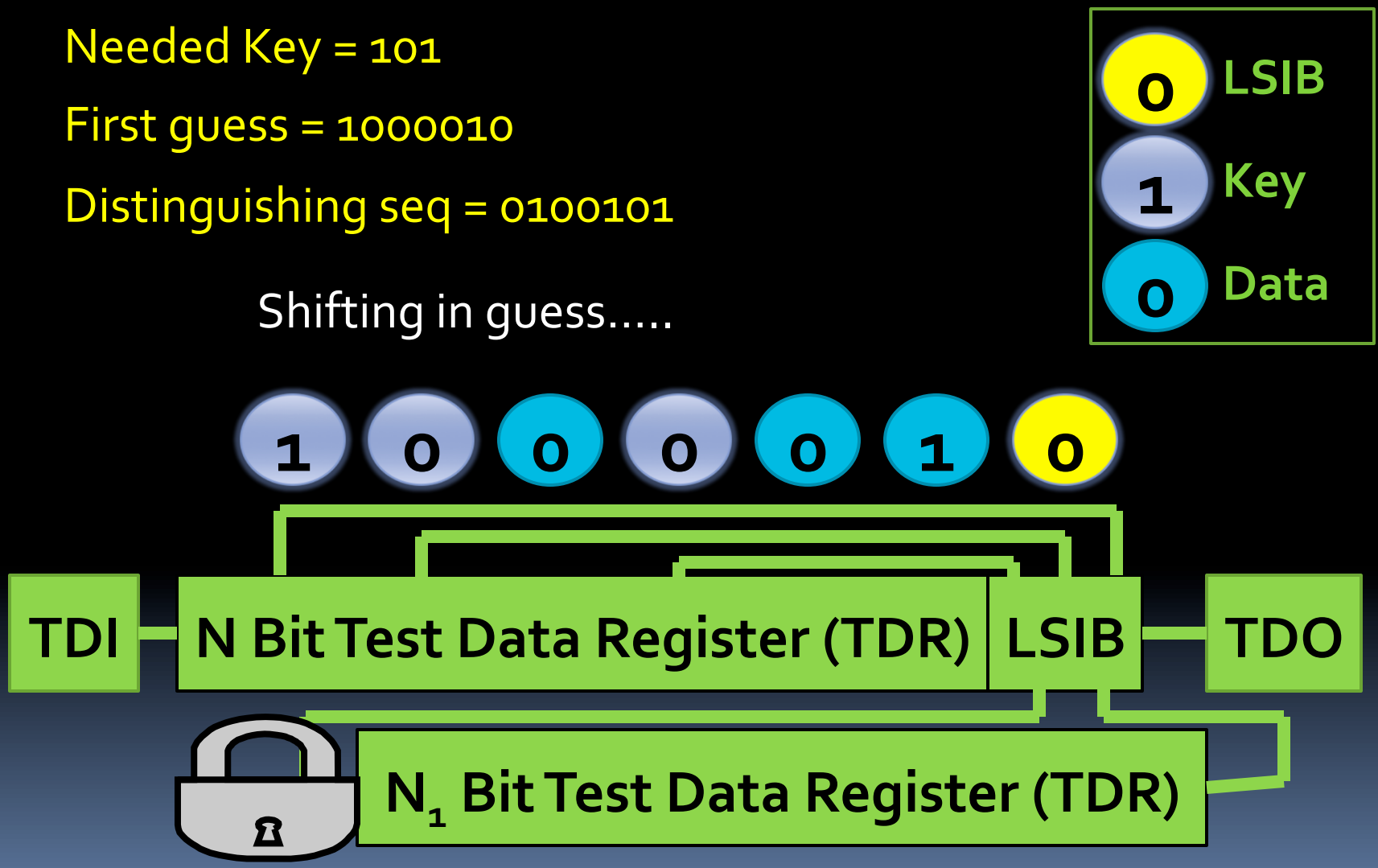
How Attackers Unlock LSIBs

Needed Key = 101

First guess = 1000010

Distinguishing seq = 0100101

Shifting in guess…..



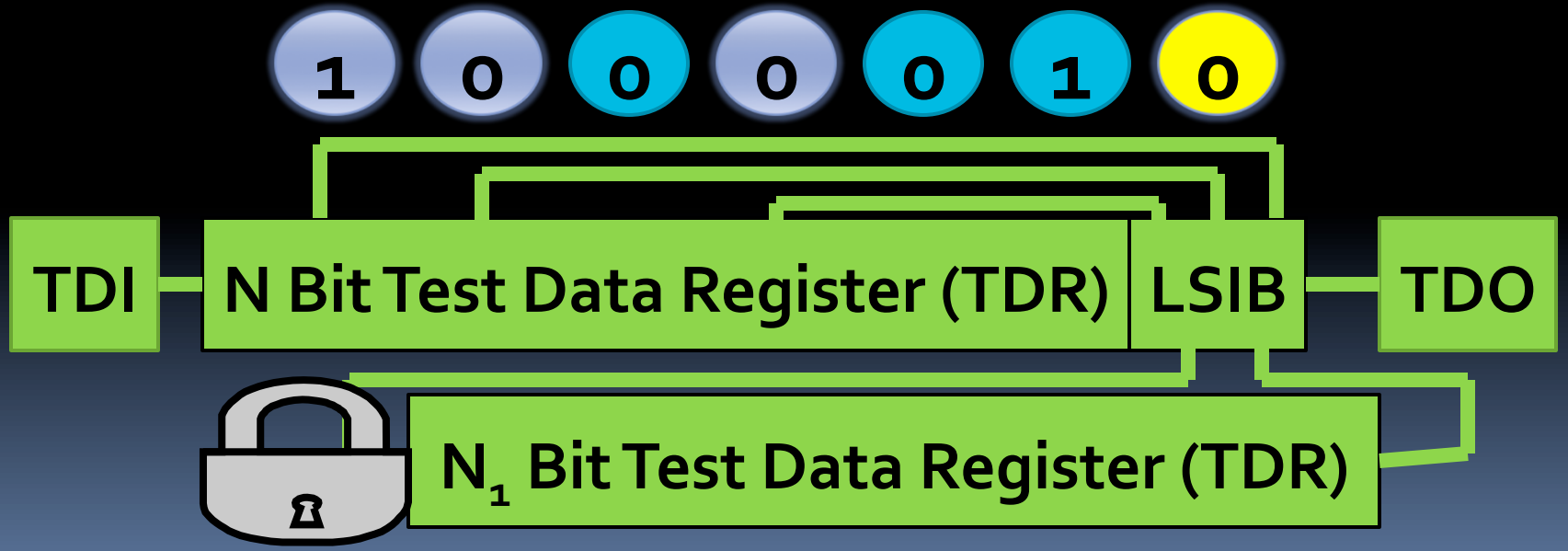
How Attackers Unlock LSIBs

Needed Key = 101

First guess = 1000010

Distinguishing seq = 0100101

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



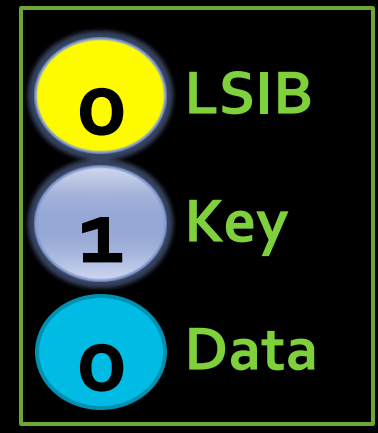
How Attackers Unlock LSIBs

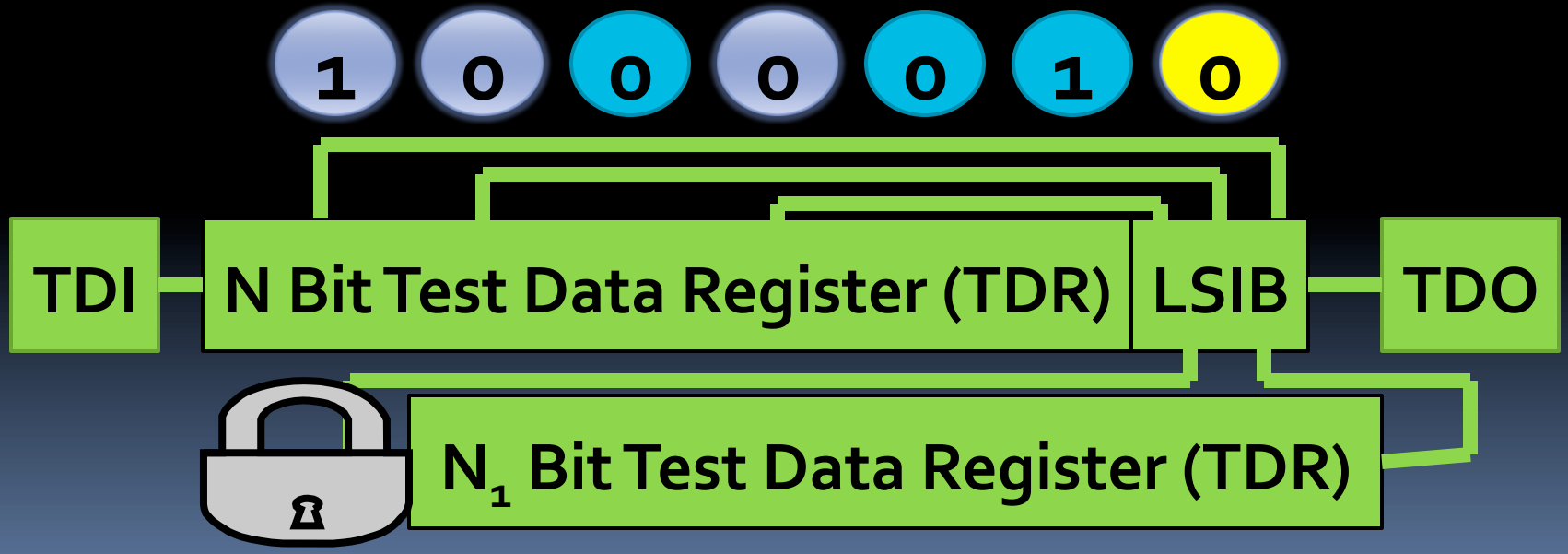
Needed Key = 101

First guess = 1000010

Distinguishing seq = 0100101

Go through RunTestIdle and Select DR





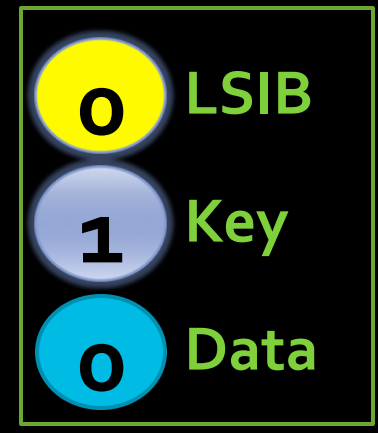
How Attackers Unlock LSIBs

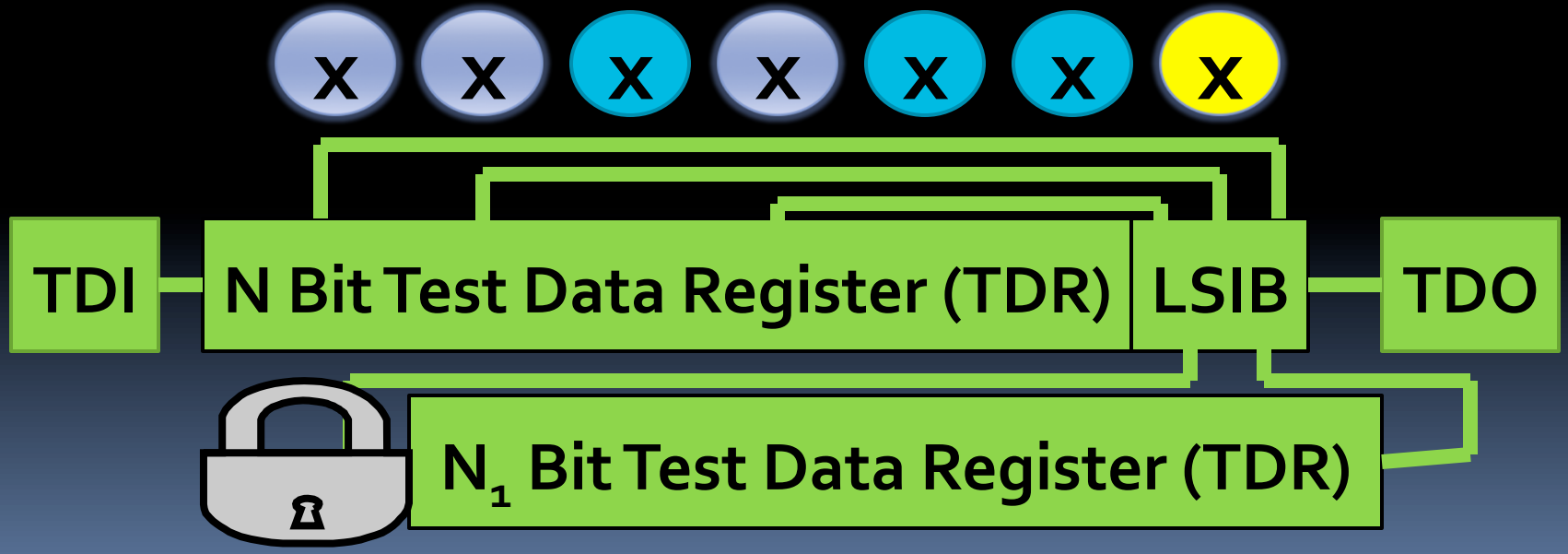
Needed Key = 101

First guess = 1000010

Distinguishing seq = 0100101

Go through CaptureDR & ShiftDR





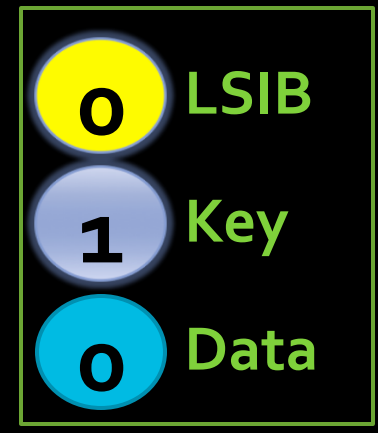
How Attackers Unlock LSIBs

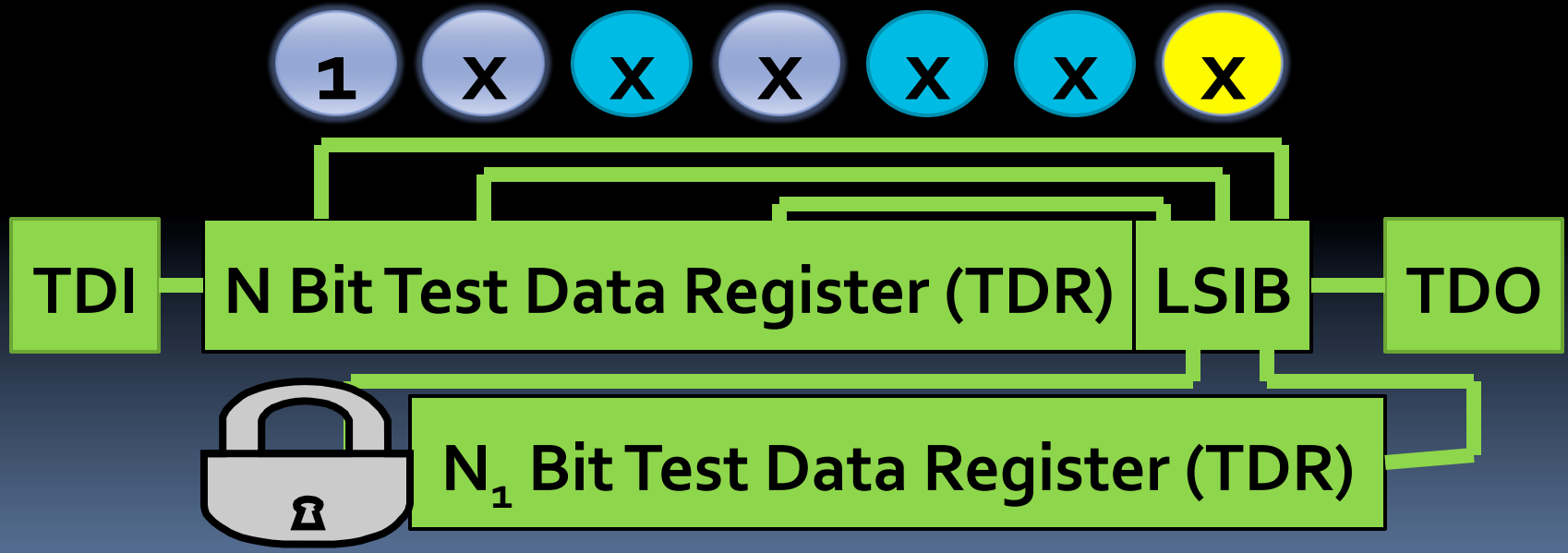
Needed Key = 101

First guess = 1000010

Distinguishing seq = 0100101

Start shifting in distinguishing sequence





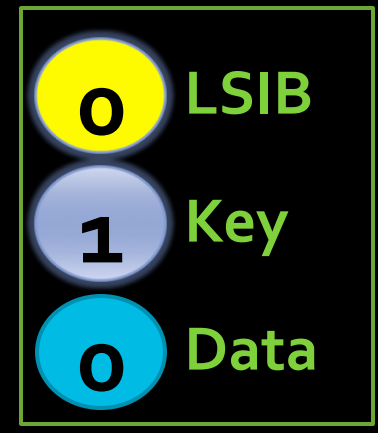
How Attackers Unlock LSIBs

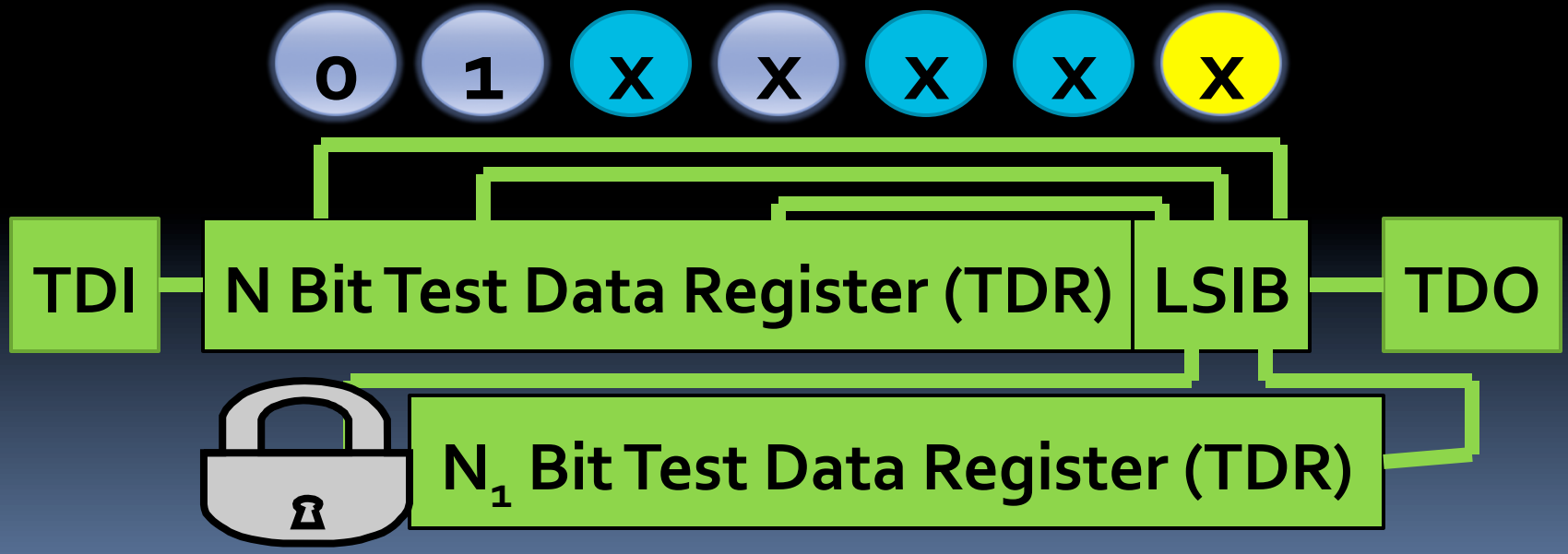
Needed Key = 101

First guess = 1000010

Distinguishing seq = 0100101

Start shifting in distinguishing sequence





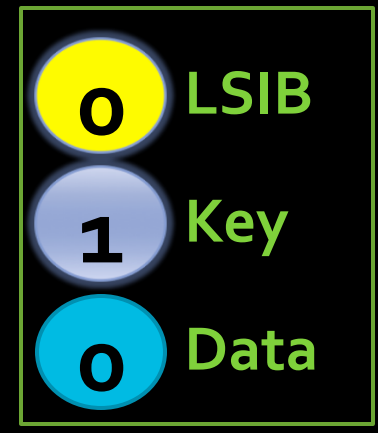
How Attackers Unlock LSIBs

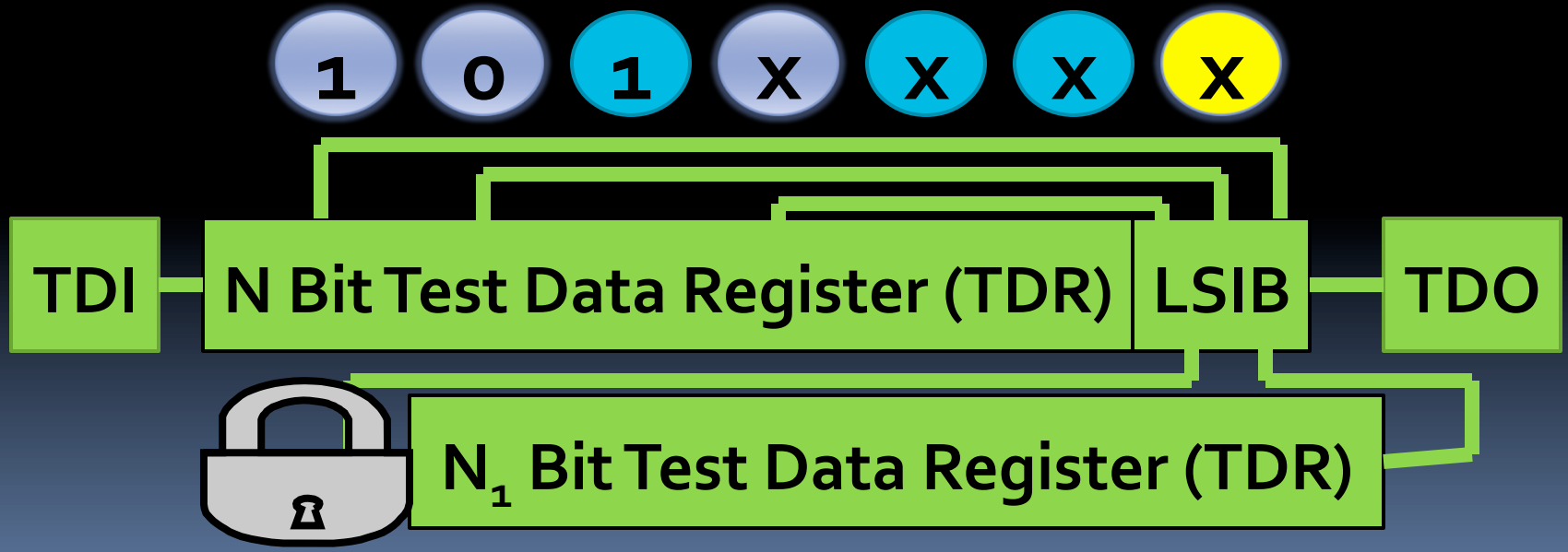
Needed Key = 101

First guess = 1000010

Distinguishing seq = 0100101

Start shifting in distinguishing sequence





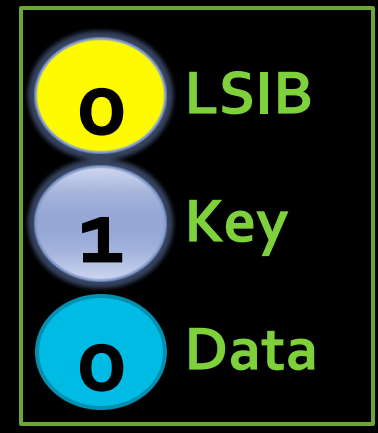
How Attackers Unlock LSIBs

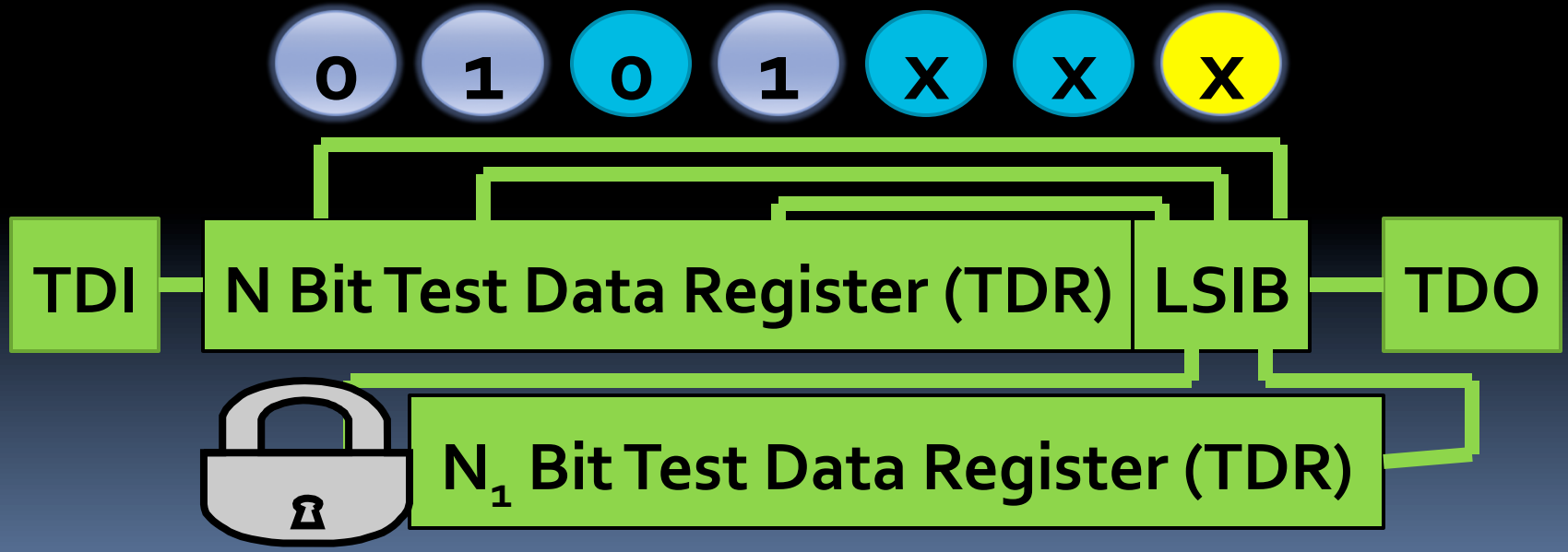
Needed Key = 101

First guess = 1000010

Distinguishing seq = 0100101

Start shifting in distinguishing sequence





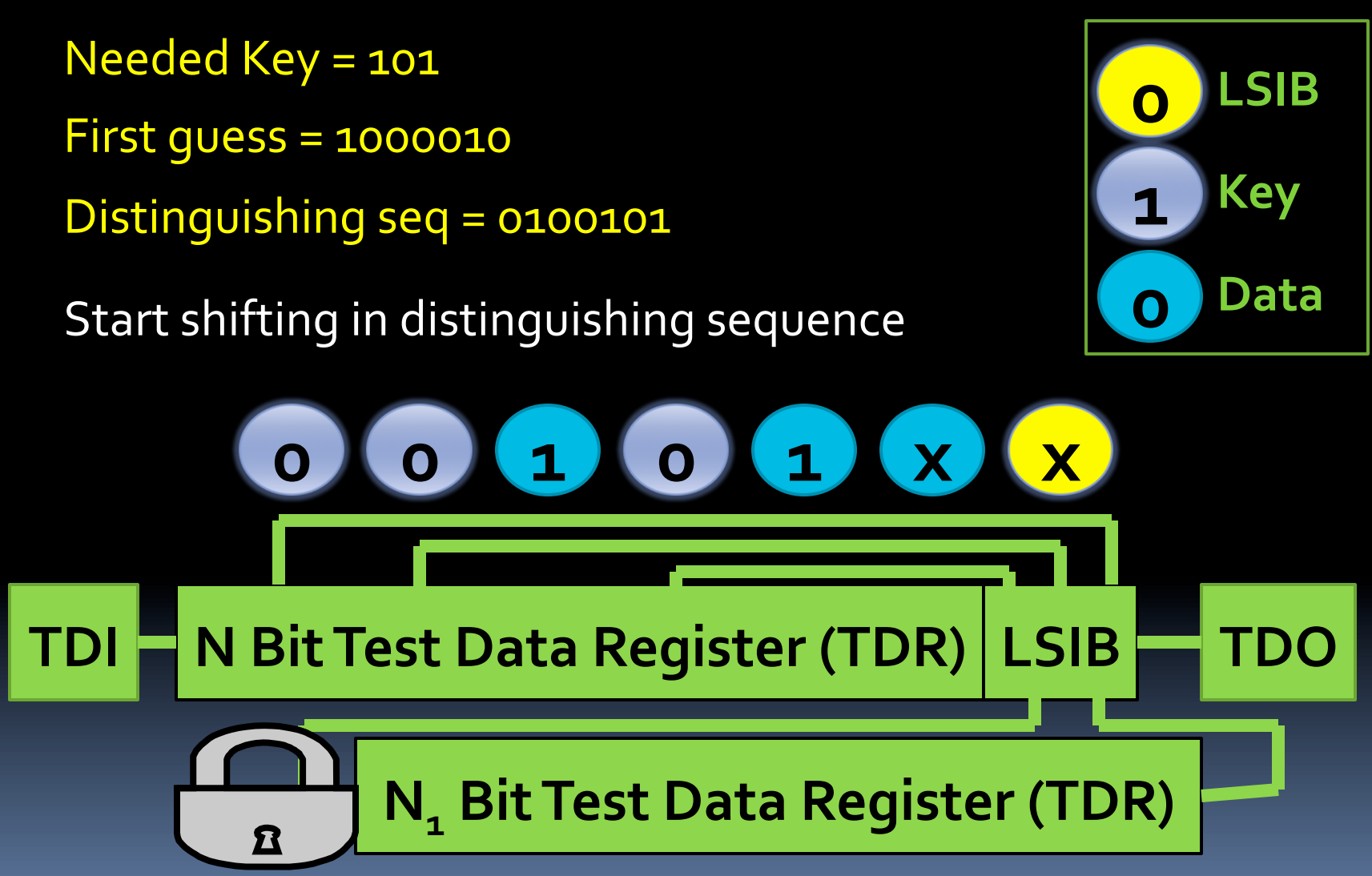
How Attackers Unlock LSIBs

Needed Key = 101

First guess = 1000010

Distinguishing seq = 0100101

Start shifting in distinguishing sequence



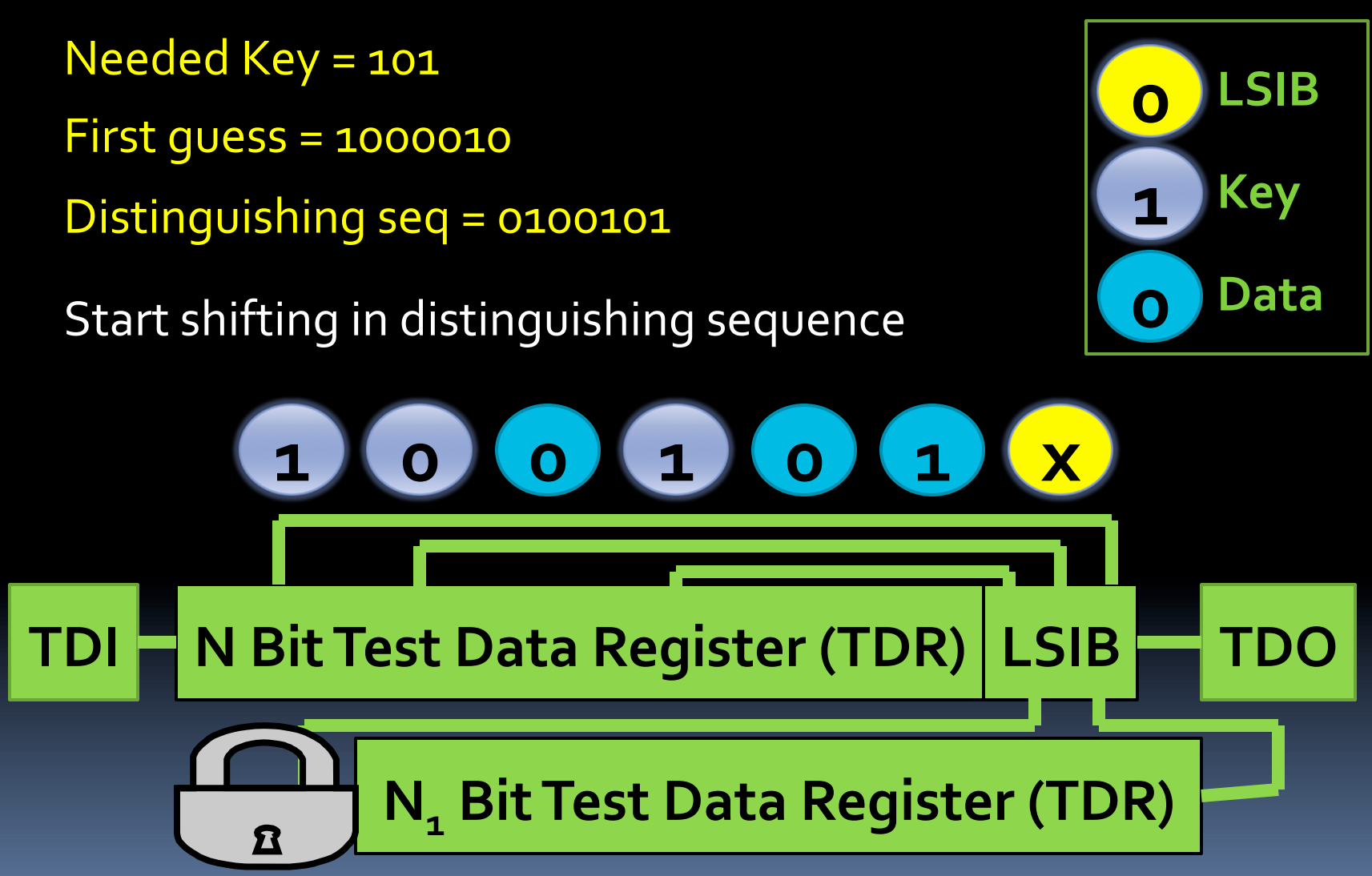
How Attackers Unlock LSIBs

Needed Key = 101

First guess = 1000010

Distinguishing seq = 0100101

Start shifting in distinguishing sequence



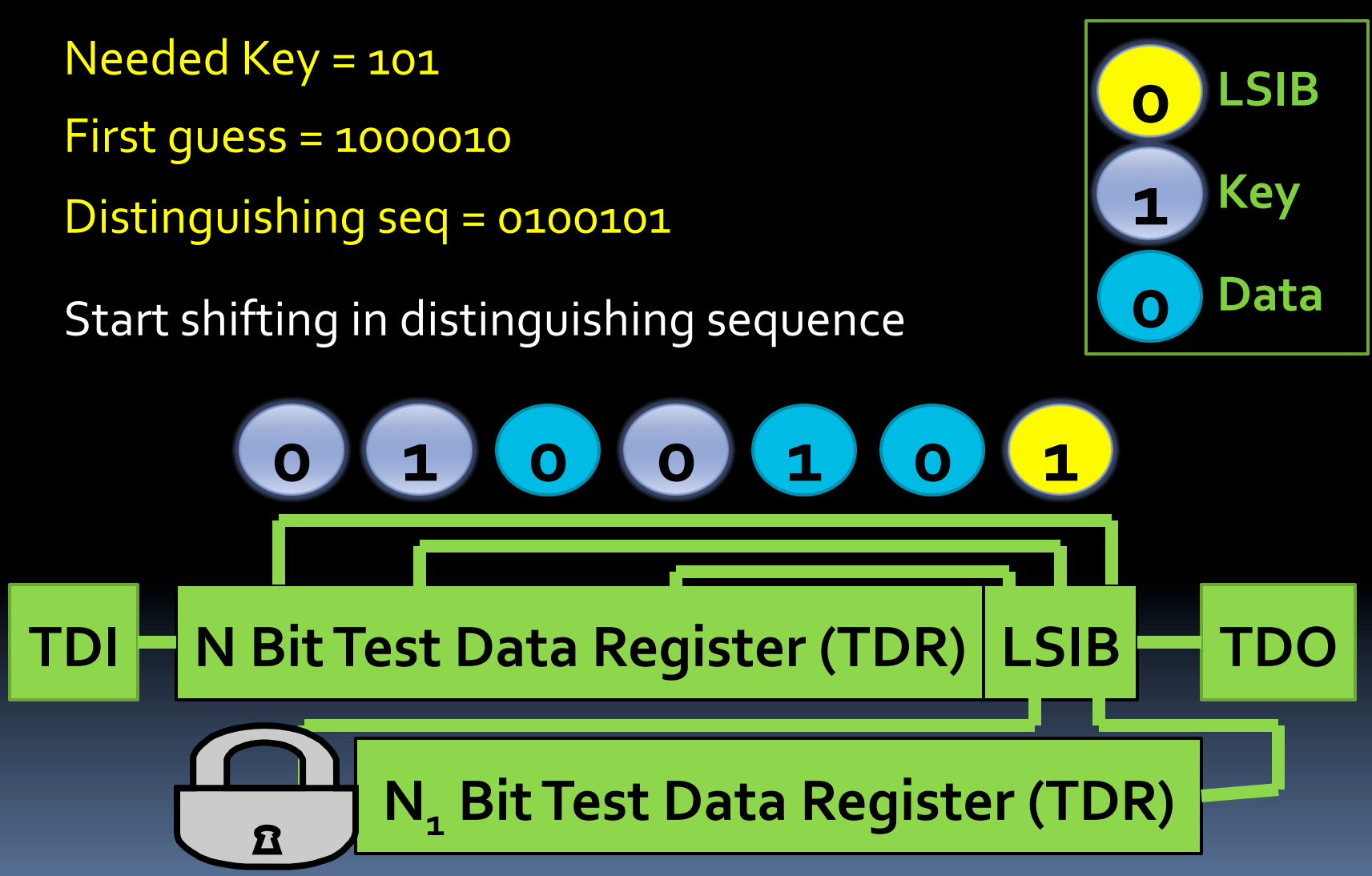
How Attackers Unlock LSIBs

Needed Key = 101

First guess = 1000010

Distinguishing seq = 0100101

Start shifting in distinguishing sequence



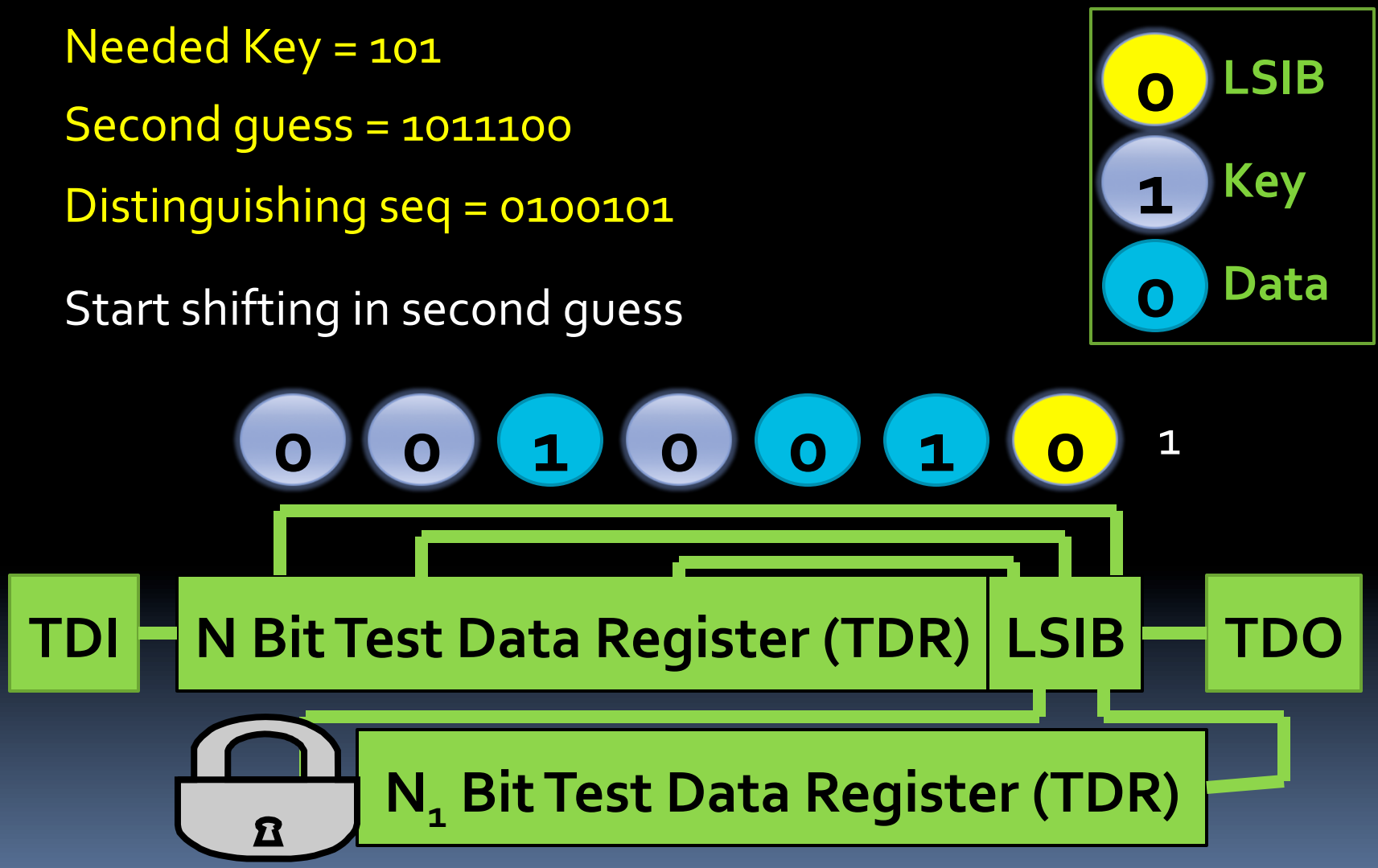
How Attackers Unlock LSIBs

Needed Key = 101

Second guess = 1011100

Distinguishing seq = 0100101

Start shifting in second guess



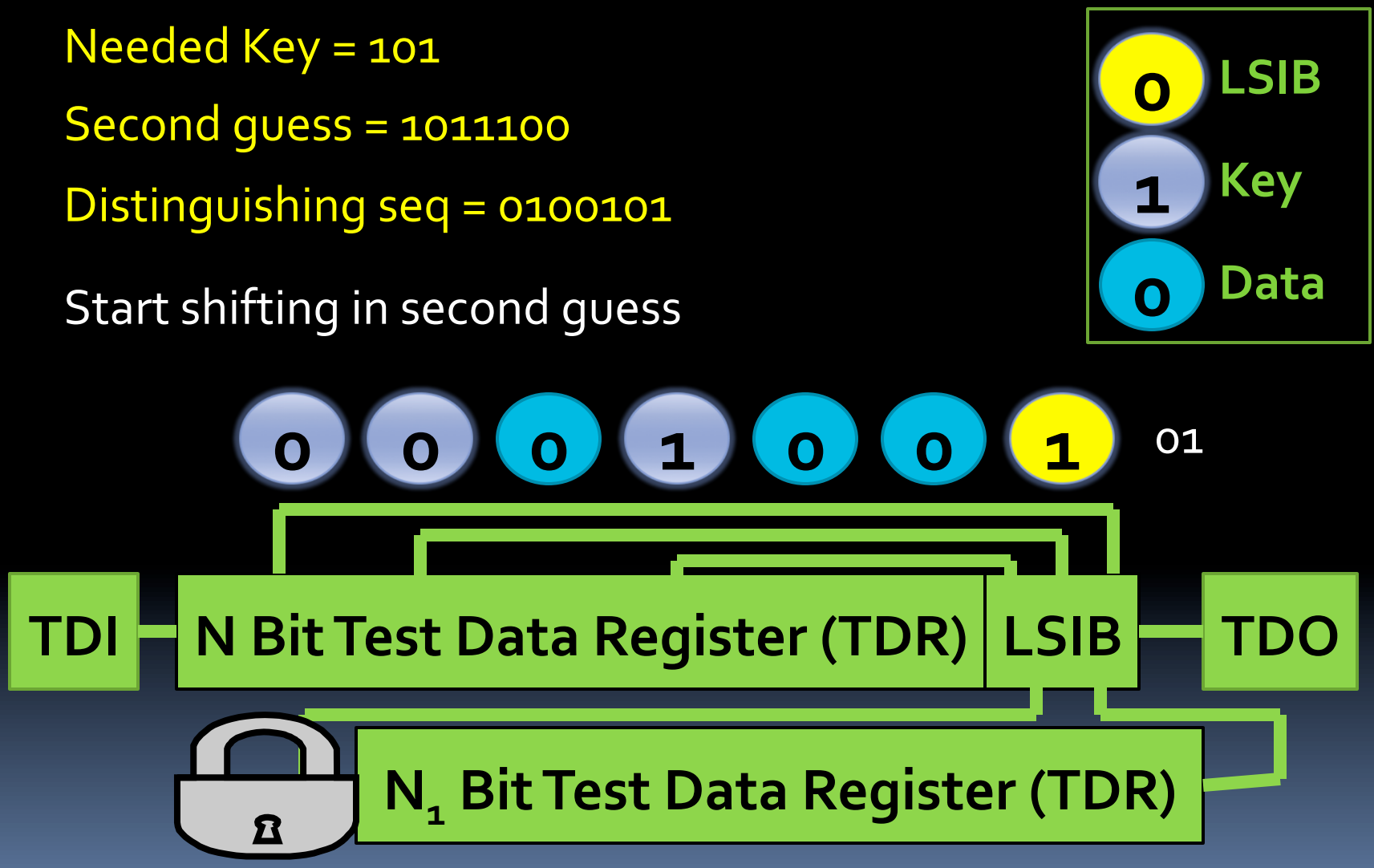
How Attackers Unlock LSIBs

Needed Key = 101

Second guess = 1011100

Distinguishing seq = 0100101

Start shifting in second guess



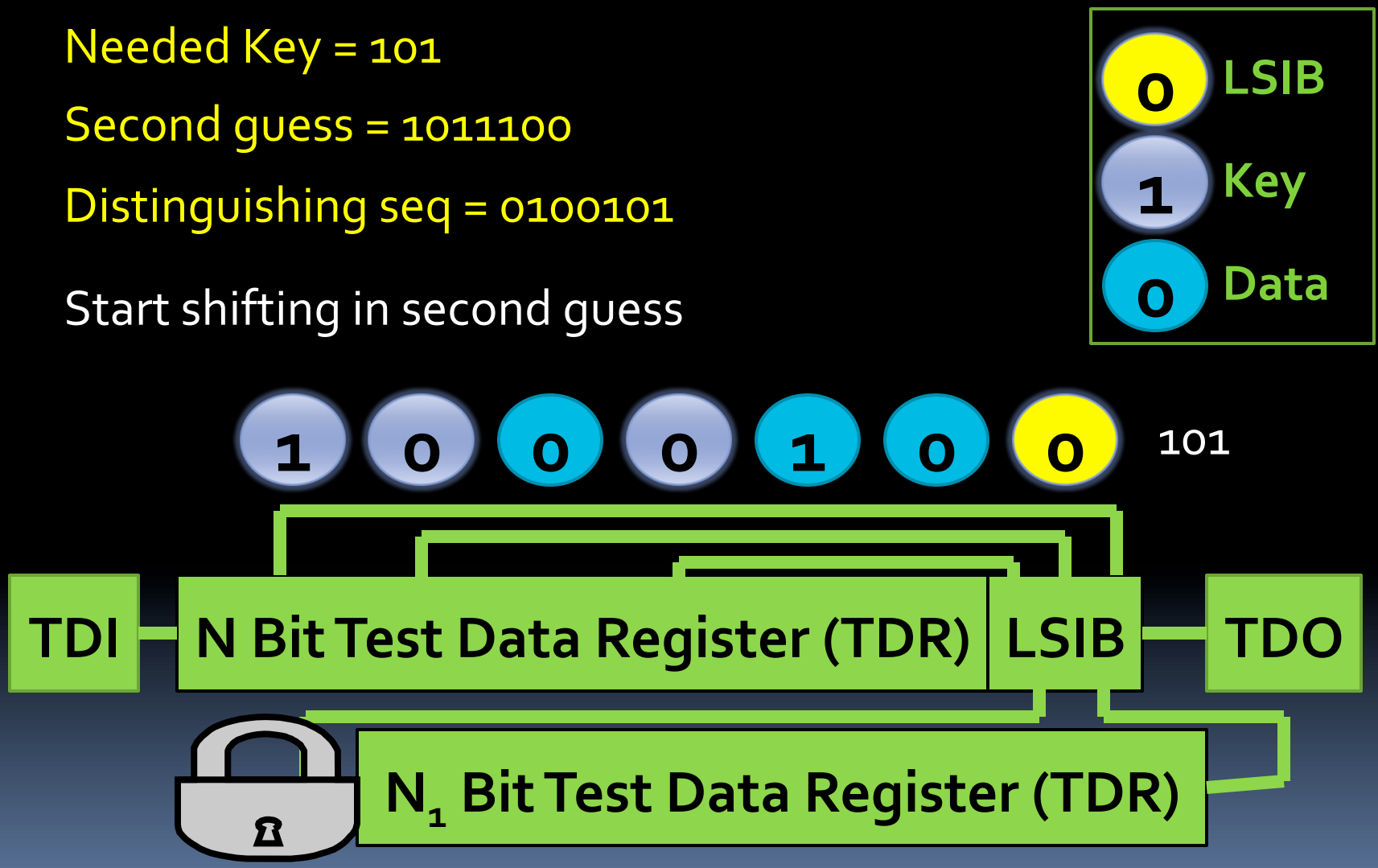
How Attackers Unlock LSIBs

Needed Key = 101

Second guess = 1011100

Distinguishing seq = 0100101

Start shifting in second guess



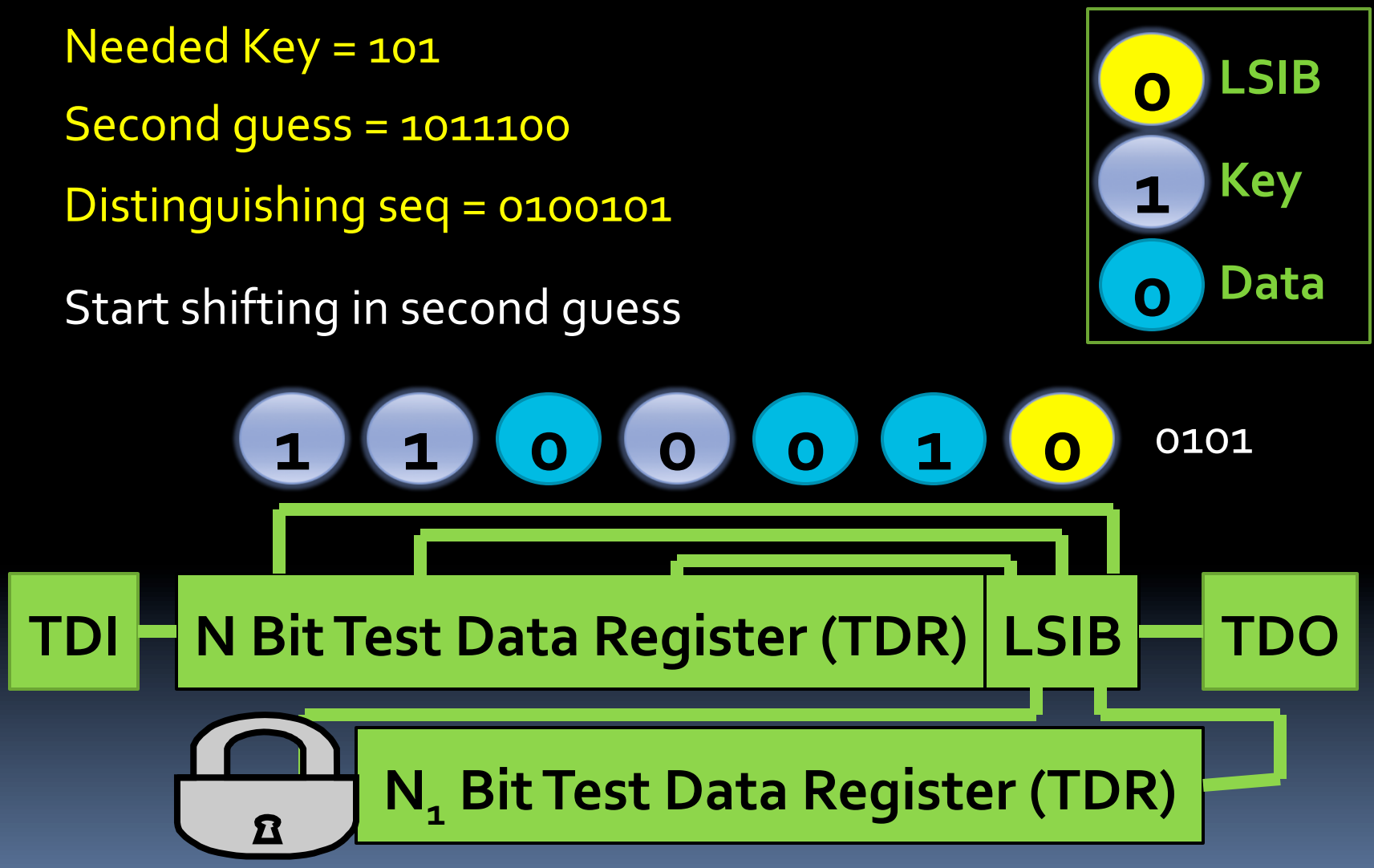
How Attackers Unlock LSIBs

Needed Key = 101

Second guess = 1011100

Distinguishing seq = 0100101

Start shifting in second guess



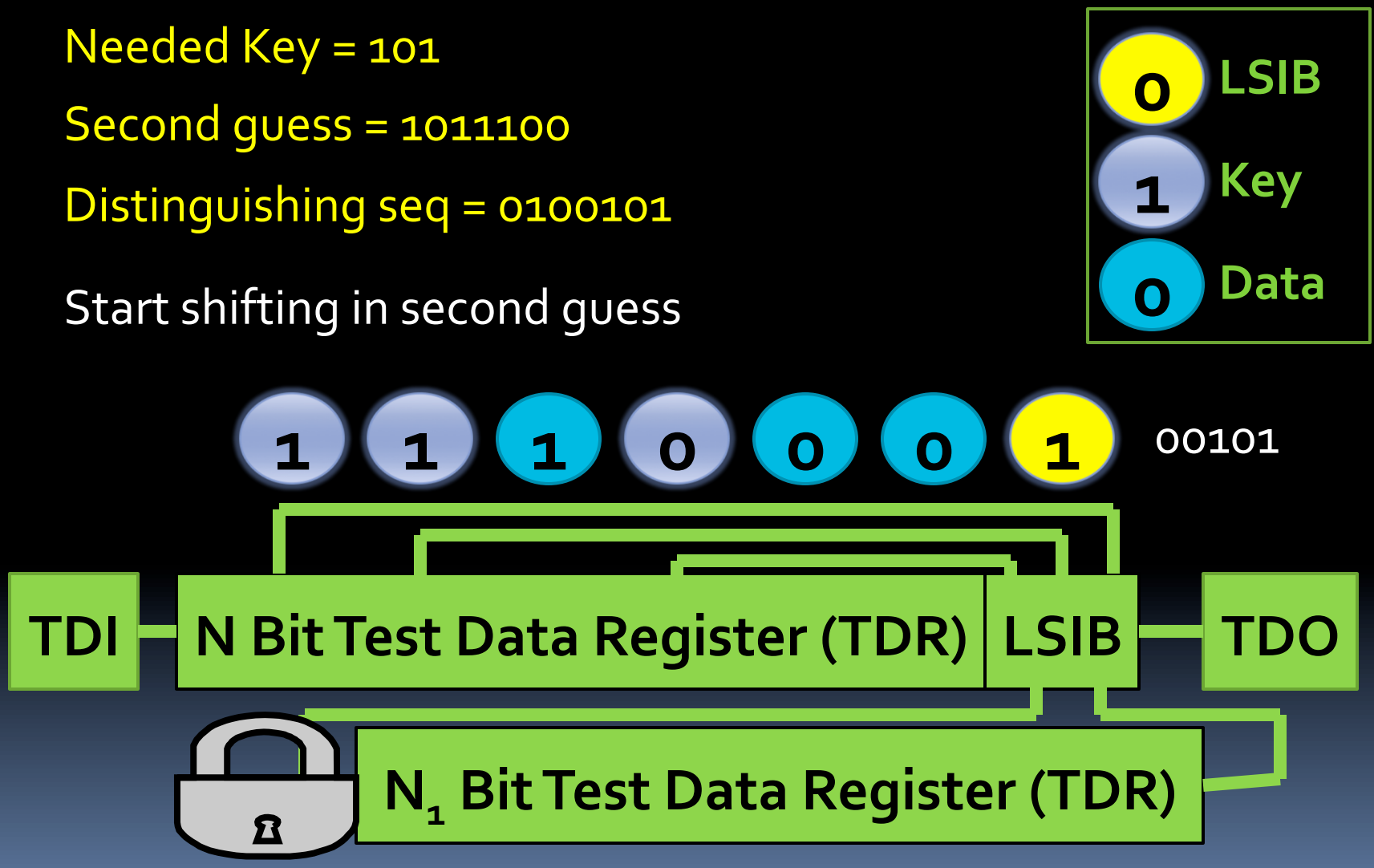
How Attackers Unlock LSIBs

Needed Key = 101

Second guess = 1011100

Distinguishing seq = 0100101

Start shifting in second guess



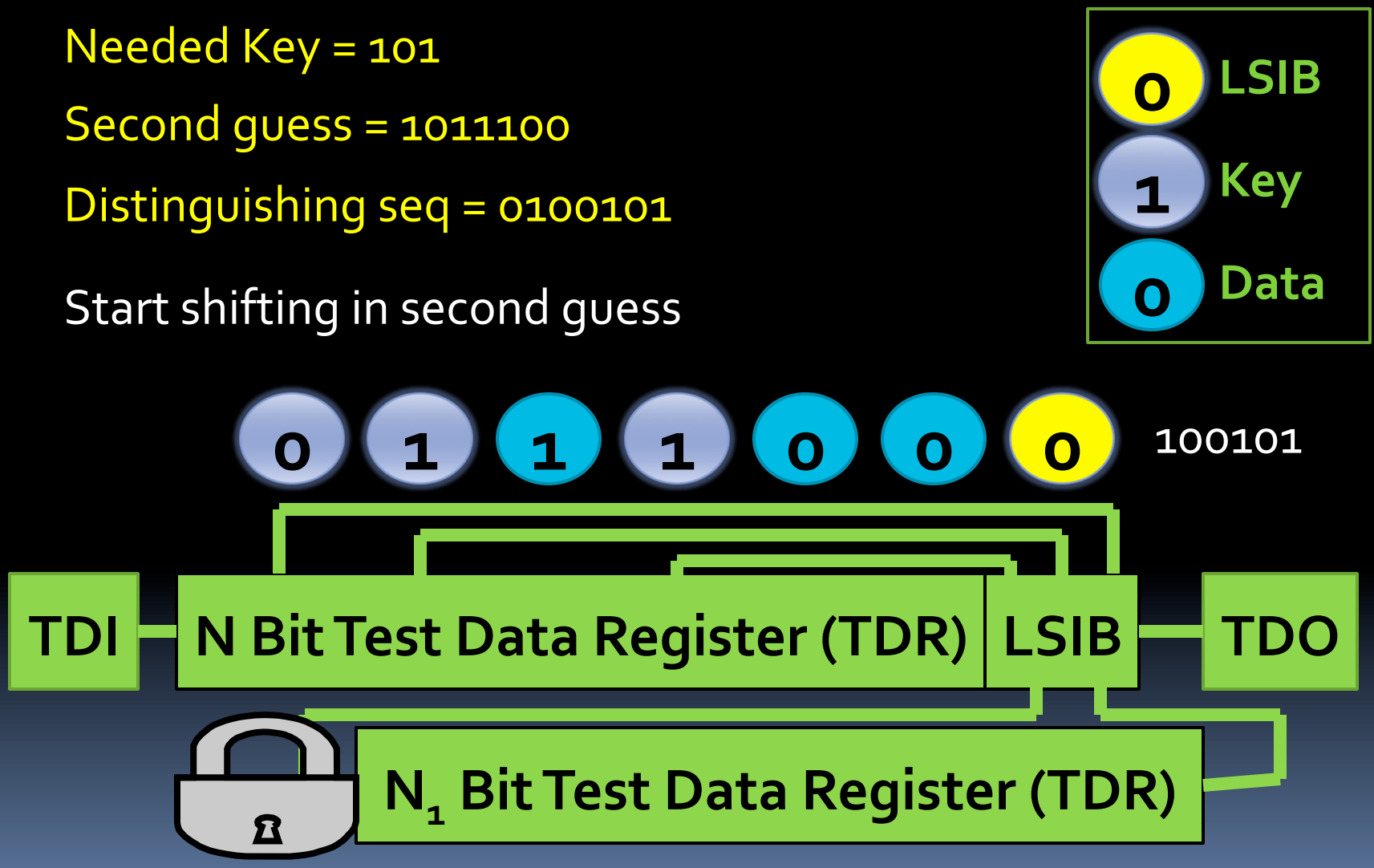
How Attackers Unlock LSIBs

Needed Key = 101

Second guess = 1011100

Distinguishing seq = 0100101

Start shifting in second guess



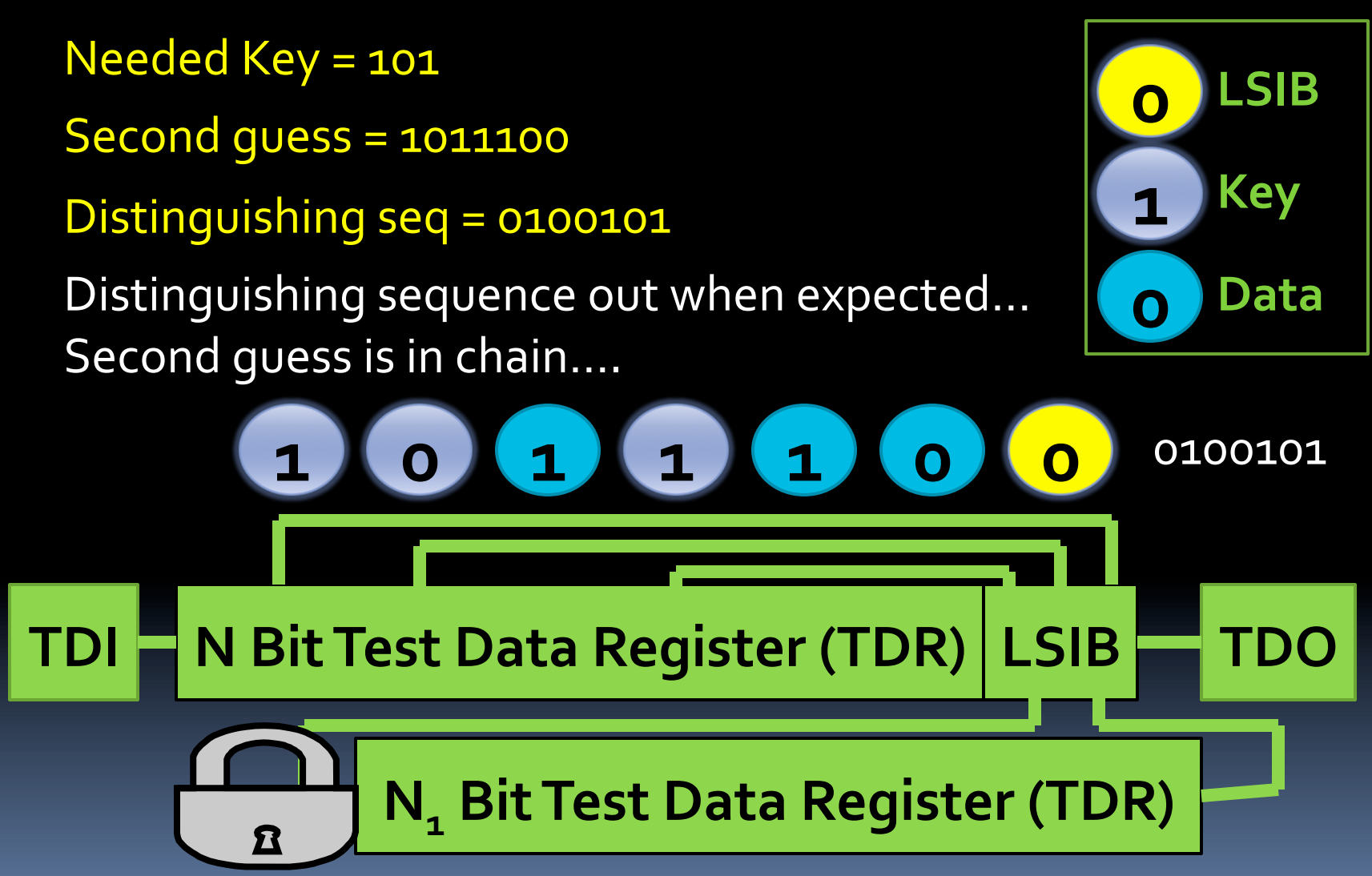
How Attackers Unlock LSIBs

Needed Key = 101

Second guess = 1011100

Distinguishing seq = 0100101

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



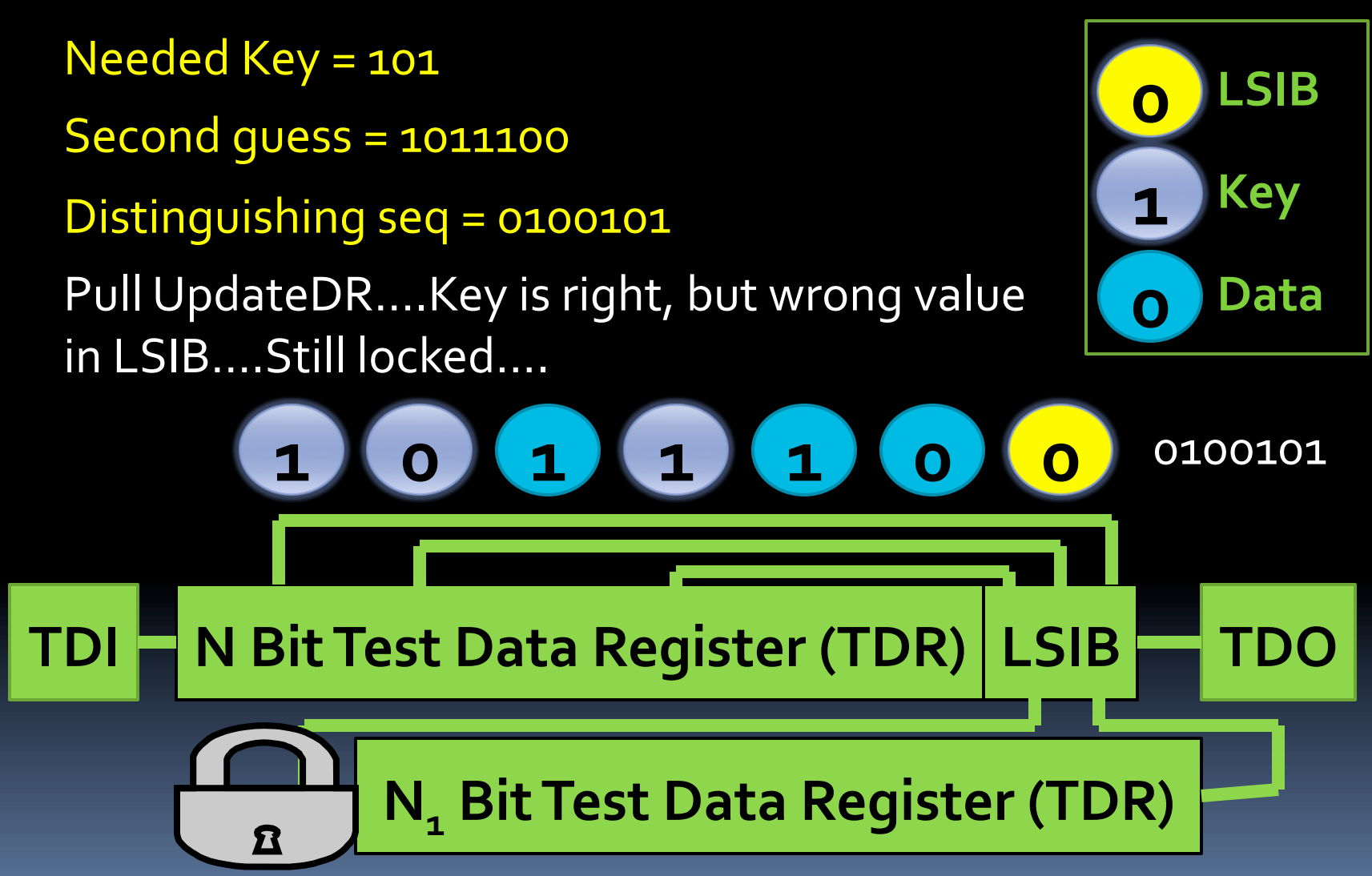
How Attackers Unlock LSIBs

Needed Key = 101

Second guess = 1011100

Distinguishing seq = 0100101

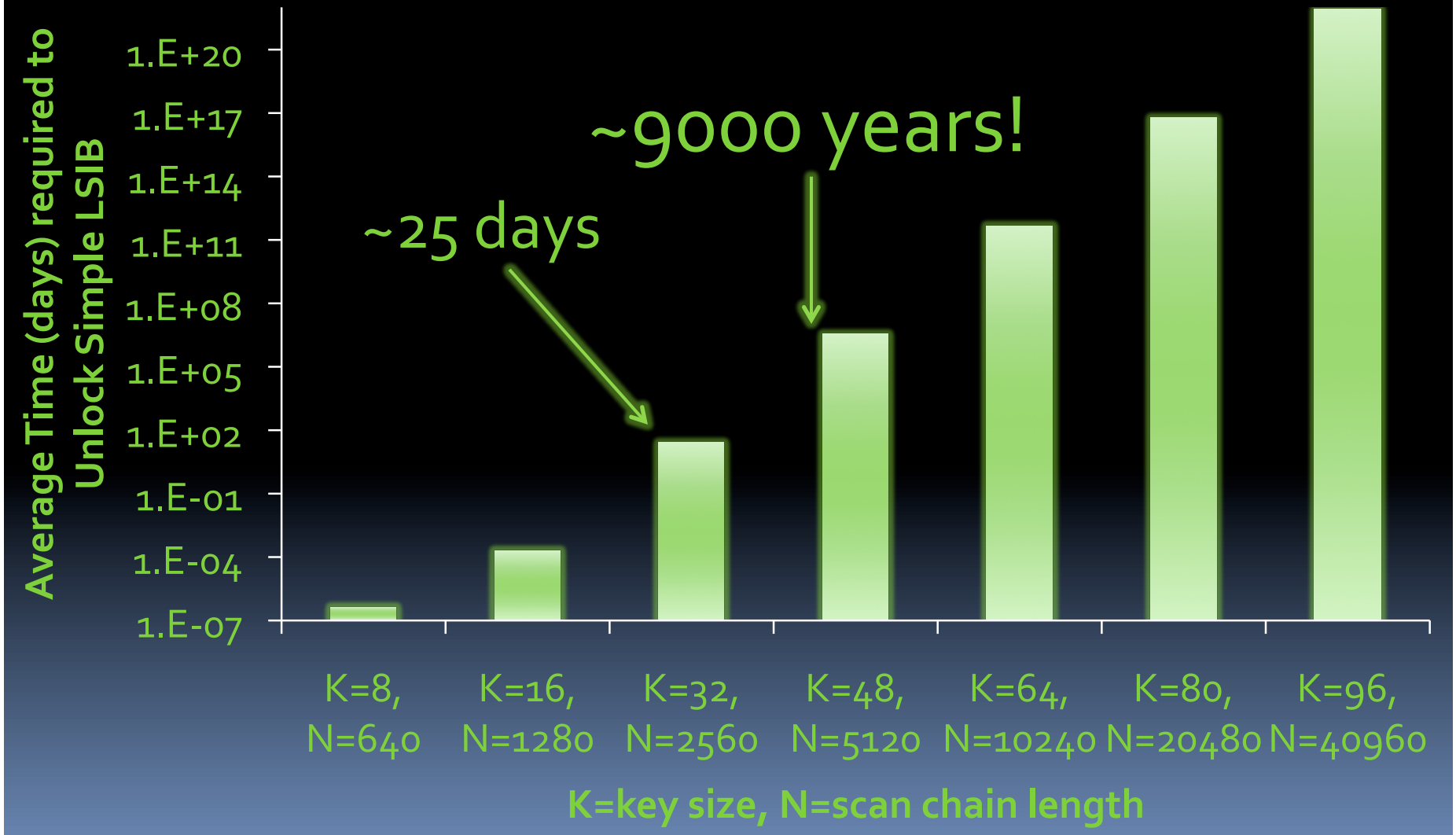
Pull UpdateDR….Key is right, but wrong value in LSIB….Still locked….



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)?



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….

Conclusions:

In an ideal world, resources would be infinite (just like the interior of the TARDIS), and we could do a perfect job of hardware security…

Conclusions

- Unfortunately, resources aren't infinite, and we need to understand security threats to make intelligent tradeoffs for design and test
- Test can both aid and hurt security unless we're careful
  - Test can help detect Trojans/Counterfeits
  - Scan chains can allow unauthorized access unless they are protected (e.g. prevent access or hide instruments)
- Some solutions are technical, while others are common sense (e.g. don't buy from unauthorized dealers).

Conclusions

It's important to consider security early in the design and test process….

If you don't even things that are easy to prevent will catch you!