Detecting Weak Keys in Manufacturing Certificates

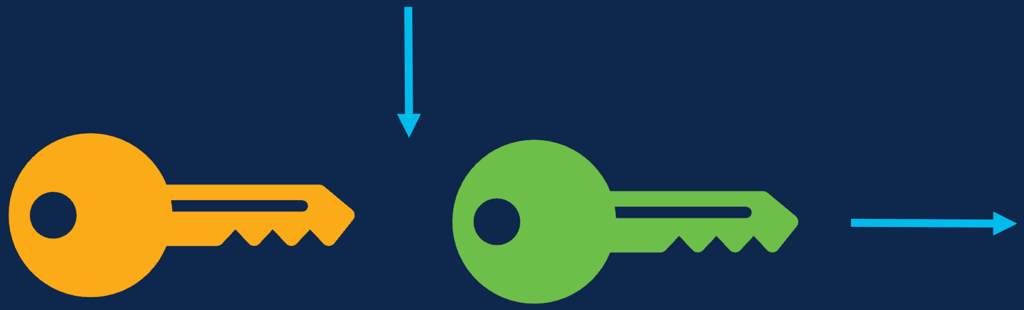
A Case Study

Andrew Chi, Brandon Enright, David McGrew

Cisco Systems

Public Keys and Certificates

Key
Generation
Algorithm



Private Key

Public Key

Certificate

Issuer
Subject
Public Key
Signature

Manufacturing (Cisco Issued),
Self-Signed, Customer Issued

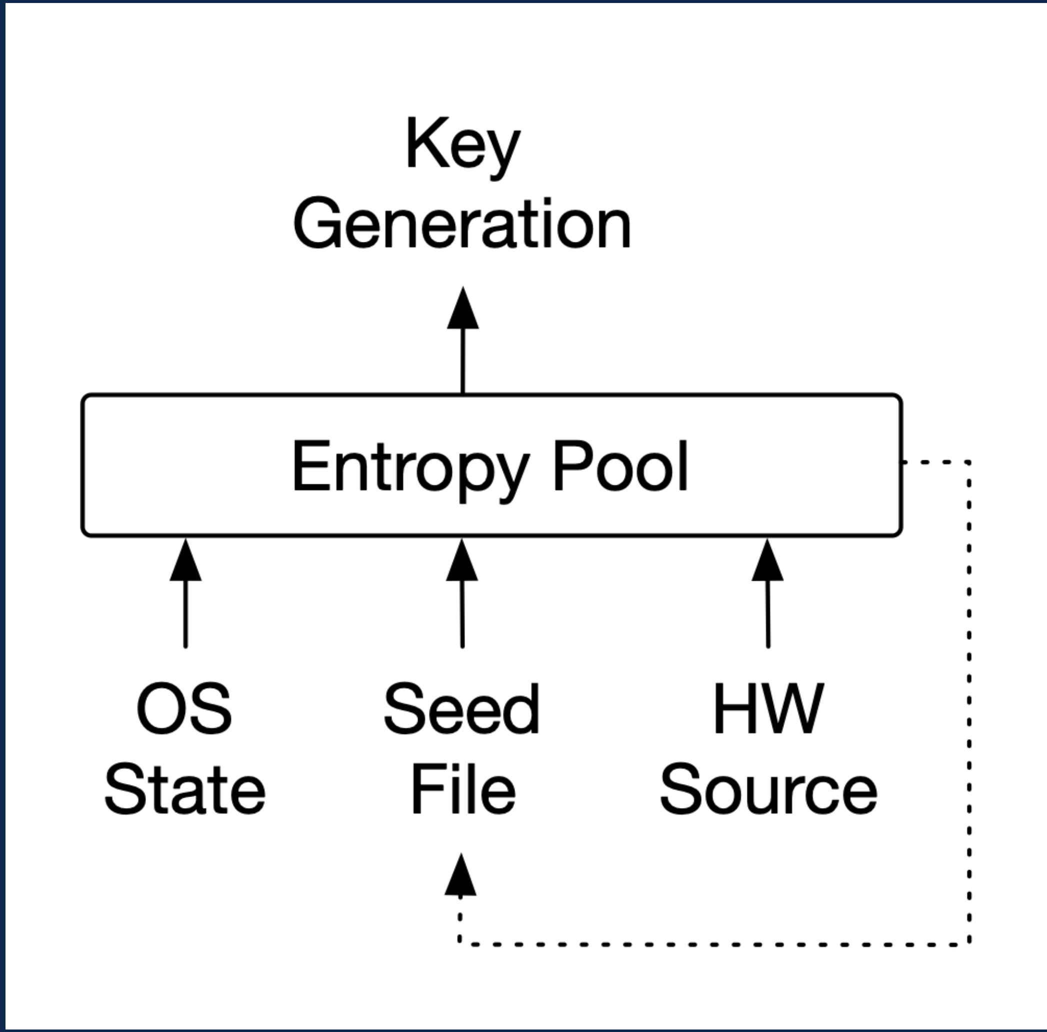
Weak Entropy: An Industry-Wide Issue

- Challenges
  - Entropy generation
  - Entropy testing
- Consistent problem
  - Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices **(Heninger, USENIX 2012)**
  - Weak Keys Remain Widespread in Network Devices **(Hastings, IMC 2016)**
  - Factoring RSA Keys in the IoT Era **(Kilgallin, IEEE TPS 2019)**
- Hard for low-end devices that generate keys immediately after bootup

Entropy and Key Generation

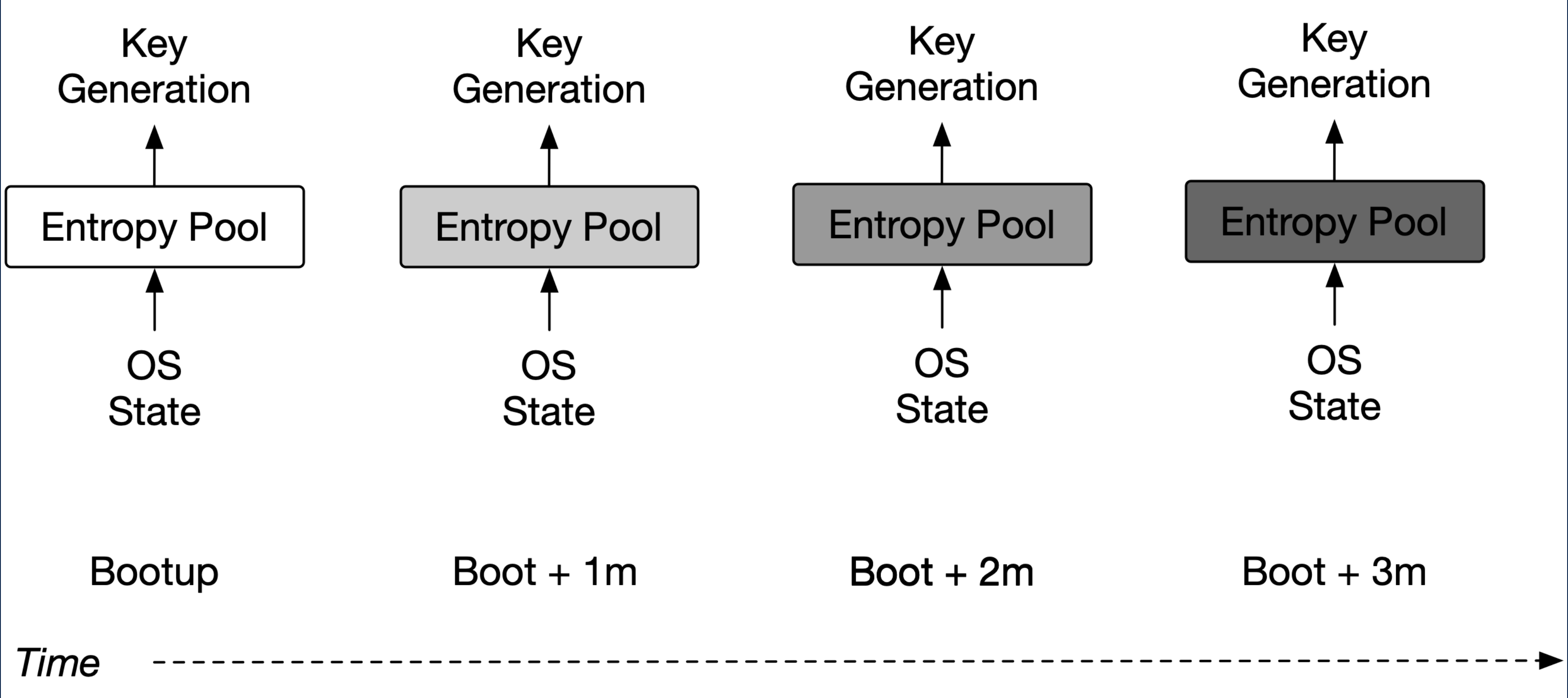
Entropy for generating keys can come from:

1. Hardware
2. Software
3. Seed file: entropy stored from previous runs



Low Initial Entropy (LIE)

Problem: Key generation right after boot, no hardware entropy



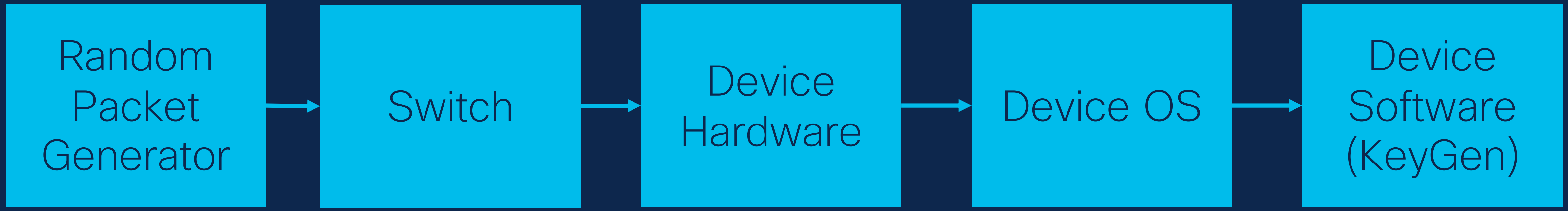
Software entropy sources accumulate unpredictability over time, and its outputs may be weak for some period after startup.

Batch Testing Can Detect Low Initial Entropy

- Testing many keys successively generated by a *single* device *will not* detect this problem
- Testing the initial keys generated by *many* devices *will* detect this problem

Good entropy requires end-to-end vigilance

- Entropy is passed through many layers
  - Failure at any stage means entropy failure
  - Components may be perfect; composition can still be flawed
  - End-to-end testing is important (unit tests insufficient)
  - Population testing is important (single-device tests insufficient)



Detection Methodology

Detection Methodology

Two main ways weak entropy shows up in certificates

Two or more RSA keys share a common factor

- Example: $n = pq$ and $m = qr$, where p, q, r are distinct primes
- Incontrovertible evidence of weak entropy
- Some products exhibit both forms of weakness (see right)

Two or more keys are identical, while the subjects (devices) are distinct

- Requires identifying devices
- Normal manufacturing practices can create thousands of certificates with identical keys and subjects
- Certificates do not always use normal 802.1AR formatting (PID, SN)
- We created custom parsing to identify devices by SN (ACT2 SN if available)

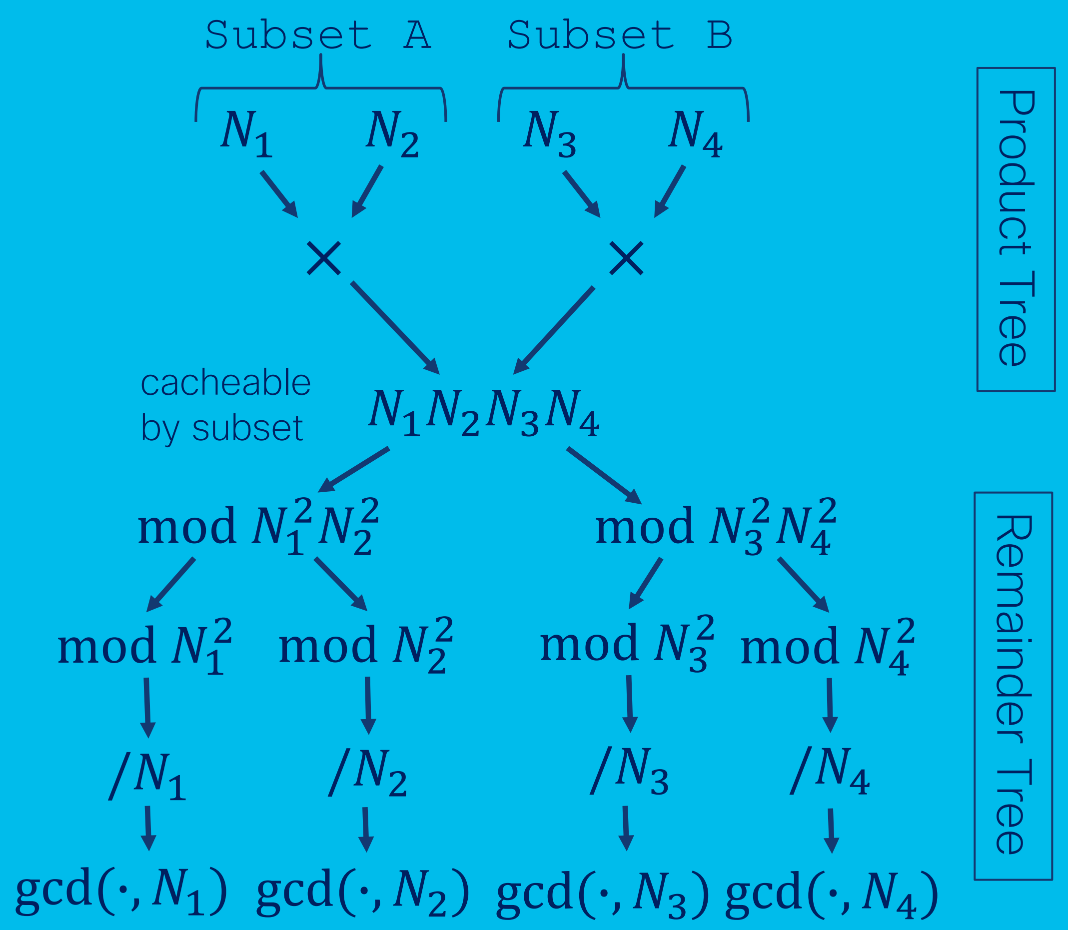
Datasets and Compute Platform

Every certificate is public data

- Dataset A: Cisco SUDI Certificates
  - Manufacturing certificates issued by Cryptographic Services
  - 200M certificates issued between 2002-2021 (mainly 2048-bit RSA keys)
- Dataset B: Public Internet Scans
  - X.509 certificates from Rapid7 Project Sonar
  - Spring 2021 only

- Compute Platform: GCP (1TB+ RAM machines)
- Goals
  - Identify RSA keys sharing a common factor
  - Identify duplicate RSA keys across distinct devices

Batch GCD



Batch GCD Algorithm (Heninger, 2012)

Suppose $n = pq$ and $m = qr$, where p, q, r are primes. Then $\gcd(n, m) = q$.

All-pairs GCD can be done in roughly $O(b \log b)$ steps, where b is the total number of bits in all of the RSA moduli.

Our Implementation

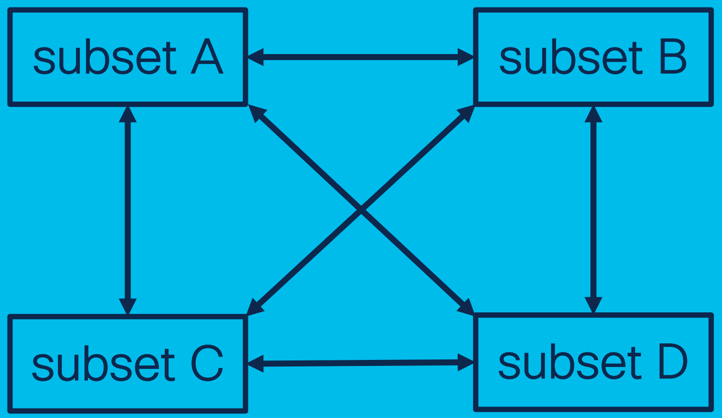
- Multithreaded
- Uses GNU Multiple Precision Arithmetic (GMP) library, unmodified

226 million certificates: larger scale than any previous work.

Cost-optimized Batch GCD

subset A	subset B	subset C	subset D

Single batch GCD over all moduli.



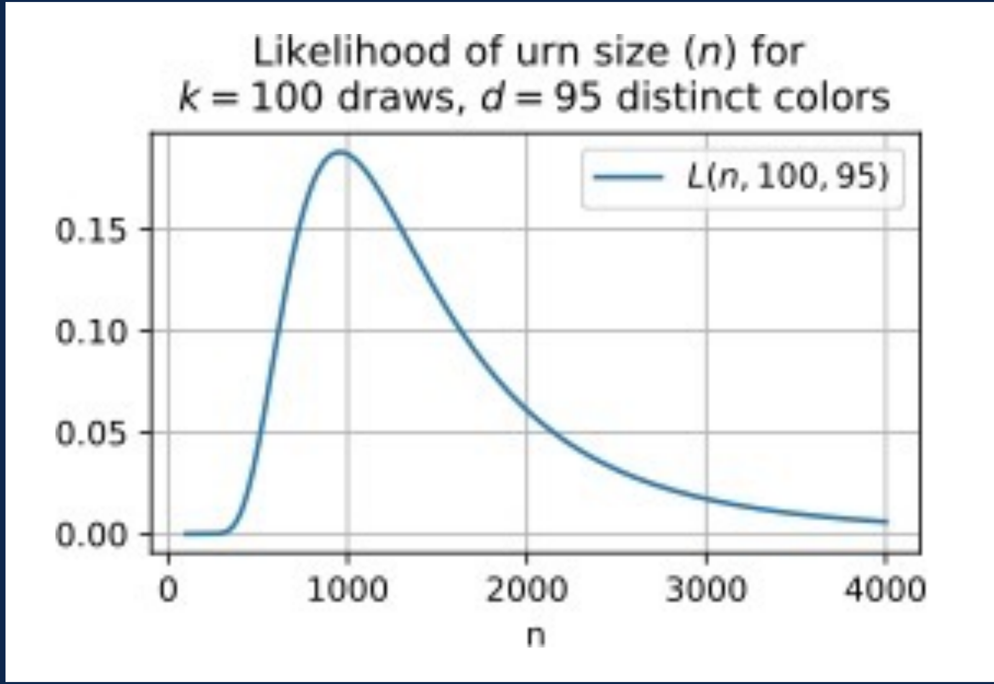
Divide moduli into subsets.
Batch GCD each pair of subsets.

	Single	Subsets
Memory	m	$m/2$
Time	t	$t/3$
Iterations	1	$\binom{4}{2} = 6$
Unit cost	k	$k/4$
Cost	kmt	$kmt/4$
Parallelizable	No	Yes

Cost can be optimized empirically based on memory and runtime measurements, GCP price structure.

Estimating the size of a weak entropy pool

- Goal: given collision(s), estimate the size of the entropy pool
- Scenario: draw k times with replacement from an urn with n balls of distinct colors; count the number of distinct colors drawn d.
- In the birthday problem, $n = 365$ and $k = 23$. About 50% chance that $d < 23$.
- Suppose we know k and d but not n; we'd like to estimate the urn size.



Likelihood vs urn size (n) for $k = 100$ draws and $d = 95$ distinct colors. The maximum likelihood estimate (MLE) for n is $\hat{n} = 957$.

Findings:
Vulnerable Cisco Products

Internal Population Test

Products with factorable or duplicate RSA keys

Current or recently supported products

PID	Product	Support Dates	Factorable	Duplicate
CP-6901	Unified IP Phone 6901	Orderable		
RV130W-A-K9-NA	RV130W VPN Router	EoSWS: 2018-08 EoHWS: 2022-08		
WS-SVC-WISM2-K9	Wireless Services Module 2	EoS: 2022-04		
AIR-CT5508-K9	5508 Wireless Controller	EoVSS: 2021-07		
AIR-CT2504-K9	2504 Wireless Controller	EoVSS: 2021-04		

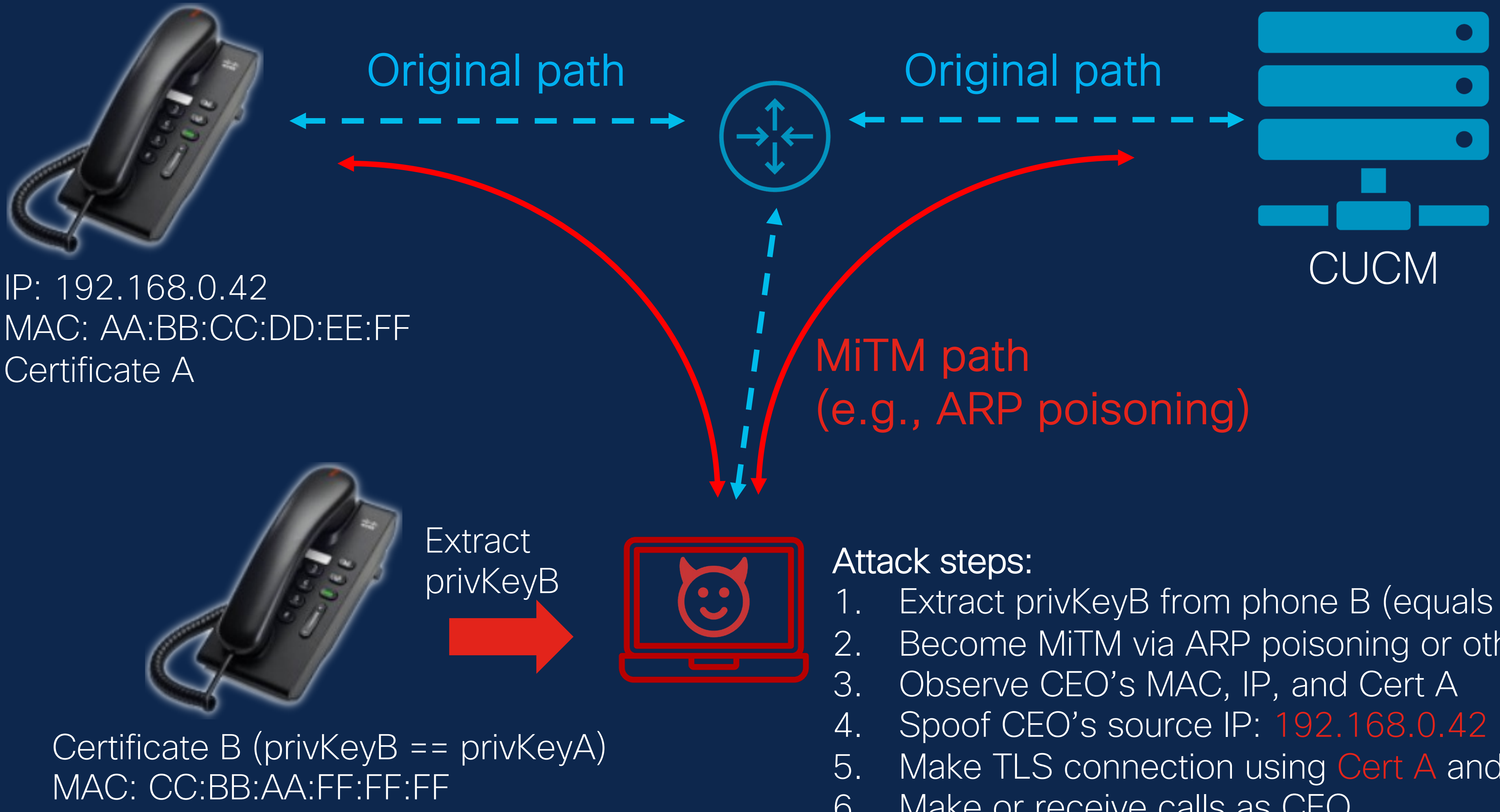
(CVE-2022-20817)

Products with factorable or duplicate RSA keys

Older EOL Products (over 4 million devices)

DMC250 (Linksys)	C1310	CP-6922	CP-8961
DMP100 (Linksys)	C1410	CP-6941	CP-9945
DMPRW1000 (Linksys)	C3201	CP-6942	CP-9951
PHM1200 (Linksys)	CP-7970	CP-6945	CP-9965
VGA2000 (Linksys)	DMC350	CP-6946	CP-9971
C1100 (Aironet)	SVR200	CP-6951	RV120W
C1130	ATA-187	CP-6961	RV220W
C1200	CIUS-7	CP-6962	
C1240	CP-6911	CP-8941	
C1250	CP-6921	CP-8945	

CEO's CP-6901 phone



Attack steps:

1. Extract privKeyB from phone B (equals privKey A)
2. Become MiTM via ARP poisoning or other method
3. Observe CEO's MAC, IP, and Cert A
4. Spoof CEO's source IP: 192.168.0.42
5. Make TLS connection using Cert A and privKey B
6. Make or receive calls as CEO

Findings: Vulnerable Internet Devices

Public Internet Scans

Frequent Strings in Factorable Certificates

Weak Certs	Issuer: Organization
1529	TPLINK | Archer
770	(unknown)
227	DrayTek Corp.
211	Cisco-Linksys, LLC
194	SonicWALL
187	Tridium
106	Netgear Inc.
74	Kronos Incorporated
64	D-Link | D-LINK
39	Cisco Small Business
32	SAMSUNG

Weak Certs	Issuer: Organization
24	Technicolor
15	Honeywell
11	Linksys International Inc.
11	Fortinet Ltd.
7	Advantech B+B SmartWorx
5	Hewlett-Packard | HP
3	Huawei
2	Gongjing
2	CalAmp Corp.
2	Primax
2	Alarm.com

- Issuers of weak certificates from Rapid7 public internet SSL scans Feb 6 – May 5, 2021

Recommendations

Recommendations

- **Prevention:** Products should use hardware entropy
  - Use HW as an entropy seed, even if implementing software crypto

- **Detection:** Run weak entropy detection tools *at scale*
  - On new certificates
  - On newly manufactured devices
  - Population testing (single-device tests insufficient)

Source code and docker image:
`https://github.com/cisco/mercury/blob/main/doc/batch-gcd.md`

Everyone matters: software, hardware, contract manufacturing

Acknowledgments

- CSIRT / Network Intelligence
  - Brandon Enright, Adam Weller, Joey Rosen, Igor Dobrogorskiy, Derek Schmell, Blake Anderson, David McGrew
- Cryptographic Services
  - Eric Hampshire, Anita Shah
- PSIRT
  - Ann Chen, Michael Schueler, Dario Ciccarone
- Product Teams
  - Shijie Zhang, Carrie Wu, Steve Yu, Karrthik Venu, Channamallikarjuna Patil, Satyanarayana Yara