Let's Remember: Cold Boot Attacks on Encryption Keys

Princeton University
Electronic Frontier Foundation
Wind River Systems

Presenter: Bo Feng

Random-access memory

- RAM is a significant form of computer data storage.
  - Dynamic RAM is a type of RAM that stores each bit of data in a separate *capacitor*. DRAM is the main memory in our PC.
  - Real *capacitor*s need to be refreshed.
- DRAM will be losing its contents when powering off….
  - Such a losing phenomenon is common and acts like a secure mechanism…
  - They will be less reliable when they are not refreshed… Also, this can be personated as a shield…
- Are these entirely true…?

Let's reconsider DRAM

- DRAMs will lose their reliability when not refreshed, however, they are not immediately erased…
  - Their contents can be saved sufficiently when they are acquired by malicious attackers…
- DRAMs will *retain their contents* after power is off.
  - To what extent can DRAM be retentive is due to the surrounding temperature…

Ice-man attack

——The topic in the paper

- Due to the limitations of DRAM, we use *cold reboots* to mount successful attacks on popular disk encryption.
- Since we use cold attack, we rely on data remanence property of DRAM to retrieve memory contents.
- We describe the extent and predictability of memory remanence.
- We develop a new algorithm to unearth cryptographic keys in memory images and to correct errors caused by bit decay.

We aim to break your disk encryption

- 1st , we try to *get* your DRAM *physically*.
- 2nd, we try to *change temperature* to slow down the decay rates.
- We have three main methods of exploiting DRAM remanence:
  - Reboot the machine and launch a custom kernel;
  - Cuts power to the machine, then restores power and boots a custom kernel;
  - Cuts the power and transplants the DRAM modules to a attacker's machine.

Avoid of "turtle speed" breaking…

- We have three methods for *reducing corruption* and for *correcting errors* :
  - *Cool the memory chips* prior to cutting power;
  - Apply our developed *algorithm* for correcting errors in private and symmetric keys;
  - *Replicate physical conditions*;
- Novel error correction algorithm:
  - Concentrates on values derive from key.

What are the motivations in this paper?

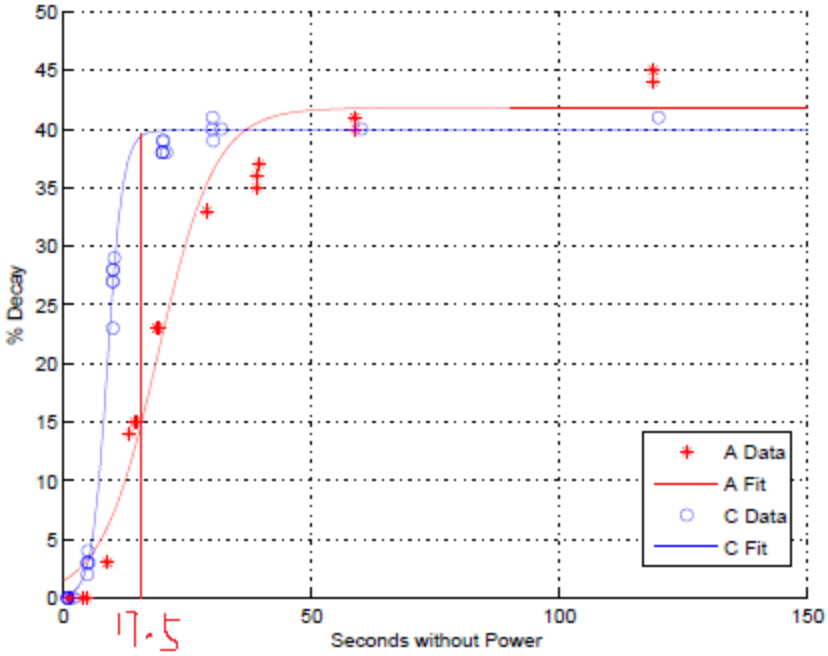
- Try to systematically prove that data in DRAM can survive reboots.
- Try to display the way to reconstruct symmetric keys in the presence of errors.
- Try to apply attacks to real disk encryption systems.
  - BitLocker
  - FileVault
  - TrueCrypt

Data Remanence Effects

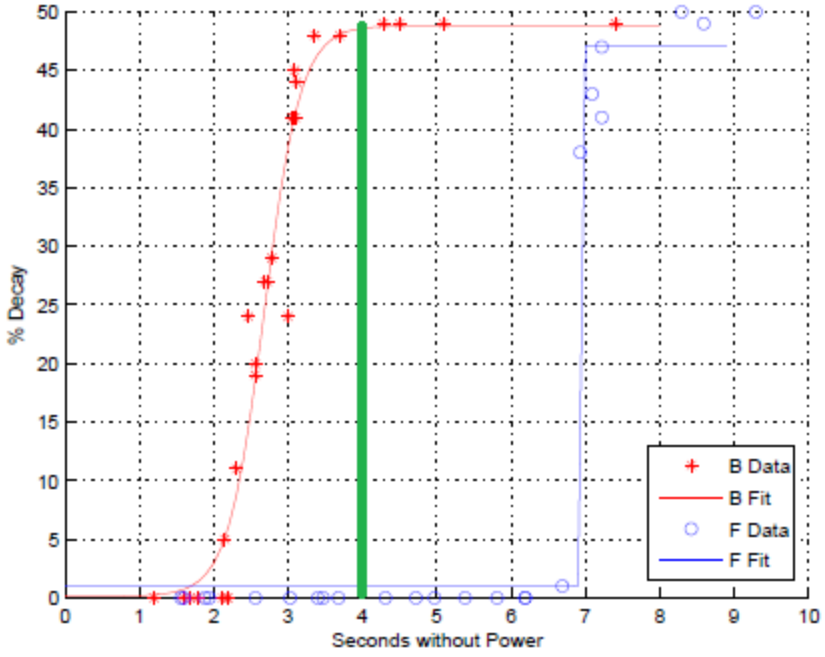
- Data remanence is the residual representation of data that remains…
- We performed trials using PC systems with different memory technologies.

	Memory Type	Chip Maker	Memory Density	Make/Model	Year
A	SDRAM	Infineon	128Mb	Dell Dimension 4100	1999
B	DDR	Samsung	512Mb	Toshiba Portégé	2001
C	DDR	Micron	256Mb	Dell Inspiron 5100	2003
D	DDR2	Infineon	512Mb	IBM T43p	2006
E	DDR2	Elpida	512Mb	IBM x60	2007
F	DDR2	Samsung	512Mb	Lenovo 3000 N100	2007

Decay at normal operating temperature

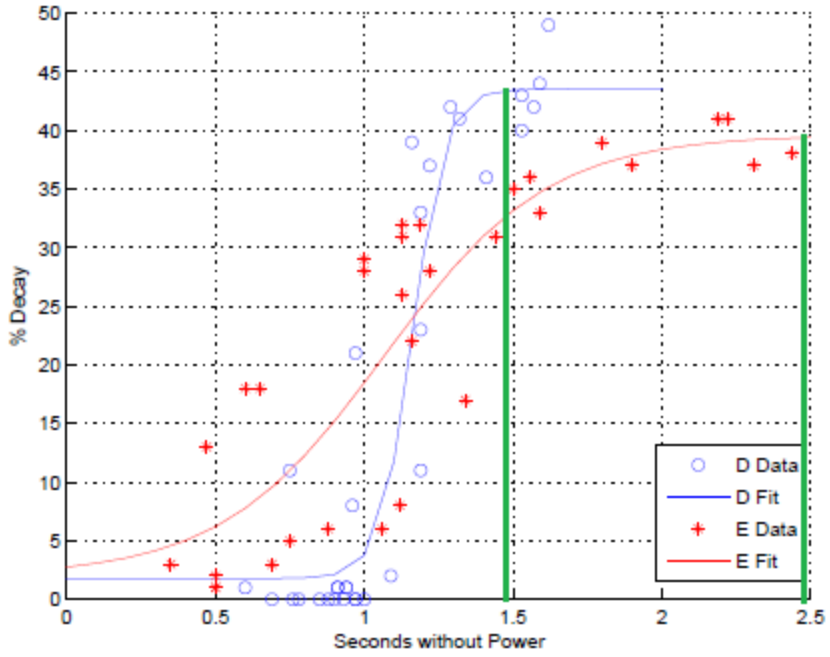


A and C



B and F

Decay at normal operating temp…



D and E

	Seconds w/o power	Error % at operating temp.	Error % at −50 °C
A	60	41	(no errors)
	300	50	0.000095
B	360	50	(no errors)
	600	50	0.000036
C	120	41	0.00105
	360	42	0.00144
D	40	50	0.025
	80	50	0.18

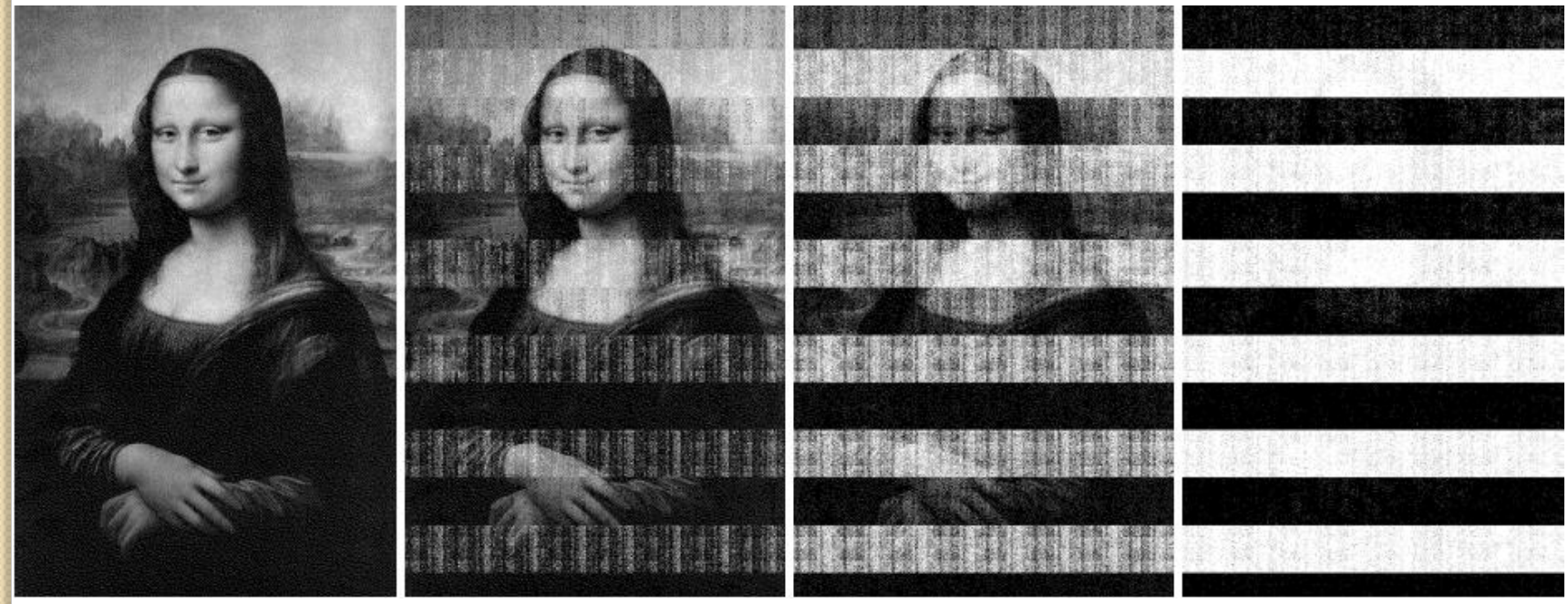
Effects of cooling on Errors

Decay at reduced temperature

- We used "*canned air*" to cool down the temperature to approximately -50° C..
  - We observed a significantly lower rate of decay…
- We used *liquid nitrogen* as an ancillary..
  - Cooled down the machines to -50° C by "canned air"
  - Immediately put the memory into liquid nitrogen..
  - Even lower decay rate…

Decay patterns and predictability

- Decay rate patterns are non-uniform…
- Decay rate patterns are predictable…



Memory region from Machine A after progressively longer intervals without power.

The picture *fades gradually* and in a *predictable* way.

Imaging Residual Memory

- Does not need special equipment…
- Our three Imaging tools:
  - PXE network boot
    - Intel's Preboot Execution Environment
  - USB drives
    - It is very common nowadays.
  - EFI boot
    - Extensible Firmware Interface

Imaging attacks combine with Imaging tools

- Three main attacks:
  - *Simple reboots*;
  - *Transferring DRAM modules*;
    - When the attacker cannot force the target system to boot imaging tools
    - Physically remove your DIMM modules

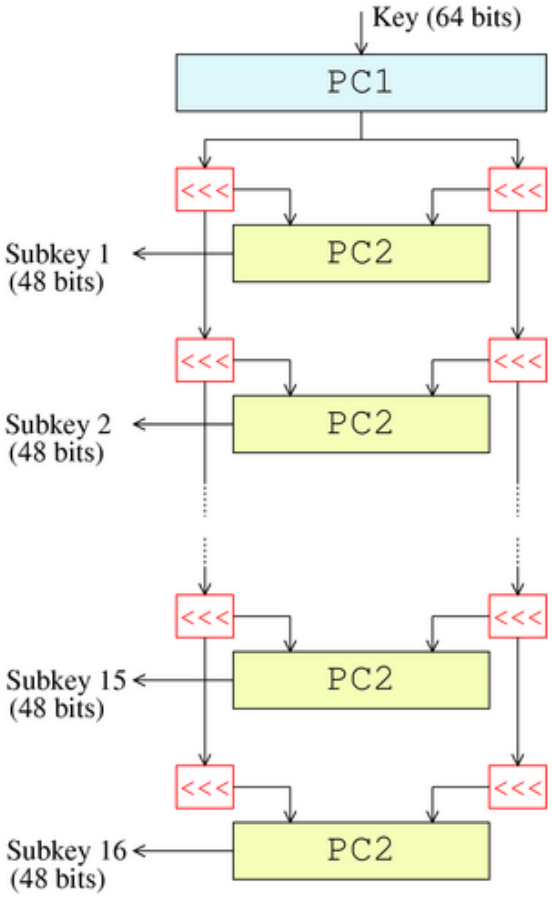


  - *Remote attacks*;
    - Use PXE tool

Key Reconstruction

- Our algorithm considers *data* other than the raw form of the key.
- Our approach to key construction has the advantage that it is completely self-contained.
- Model the decay:
  - All memory bits tend to decay to *predictable ground states*.
  - Only a tiny fraction flipping in the opposite direction.

Reconstructing DES keys



- This technique can be extended to correct errors in Triple DES keys.

Reconstructing AES keys

- AES key schedule is more complex than DES key schedule.
- *We search keys in order of distance to the recovered key and output any key whose schedule is sufficiently close to the recovered schedule.*
- We optimize the algorithm by taking advantage of AES key schedule's structure.
- Reconstructing *Tweak keys* and *RSA keys* can reference to the paper…

Identifying keys in memory

- We have developed fully automatic techniques for locating symmetric encryption keys in memory images.
- We target the key schedule instead of the key itself.
- Previous approaches have other practical drawbacks:
  - High false positives;
  - They are not robust to memory errors.

Identifying AES keys

- The algorithm we propose:
  - Iterate through each byte of memory. Treat the following block of 176 or 240 bytes of memory as an AES key schedule;
  - For each word in the potential key schedule, calculate the Hamming distance from that word to the key schedule word that should have been generated from the surrounding words.
  - If the total number of bits violating the constraints on a correct AES key schedule is sufficiently small, output the key.

Identifying RSA keys

- The format for an RSA private key we use is specified here: ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-1/pkcs-1v2-1.pdf
- Such format suggests two techniques:
  - Search for known *contents* of the fields;
  - Look for memory that matches the *structure* of the DER encoding.

Attacking encrypted disks

- Disk encryption is widely used nowadays…
- Disk encryption gives most computer users misconceptions…
- Most full disk encryption schemes are vulnerable…
- In this paper, we have applied some of the tools to attack popular disk encryption systems.
  - The most time consuming part:
    - setting up the encrypted disks and verifying that we had actually found the correct decryption keys.

Attacking BitLocker

- BitLocker operates as a filter driver that resides between the file system and the disk driver, encrypting and decrypting individual sectors on demand.
- BitUnlocker---externel USB hard disk containing Linux

Attacking the FileVault

- We used EFI memory imaging program to extract a memory image from an Macintosh system with a FileVault volume mounted.
- "keyfind" automatically identified the FileVault AES keys, which did not contain any bit errors in the tests.

Attacking the TrueCrypt

- **TrueCrypt** is a software application used for on-the-fly encryption. It can create a virtual encrypted disk within a file or encrypt a partition.
- Individual algorithms supported by TrueCrypt are AES, Serpent and Twofish. Additionally, five different combinations of cascaded algorithms are available: AES-Twofish, AES-Twofish-Serpent, Serpent-AES, Serpent-Twofish-AES and Twofish-Serpent. The cryptographic hash functions used by TrueCrypt are RIPEMD-160, SHA-512 and Whirlpool.

Countermeasures and Limitations

- Memory imaging attacks are difficult to defend against. (cryptographic keys have to be stored somewhere…)
- Discarding or obscuring cryptographic keys, physically protecting DRAM chips, making the contents of memory decay more readily…

Continue…

- Scrubbing Memory;
- Limiting booting from network or removable media;
- Suspending a system safely;
- Avoiding precomputation;
- Key expansion;
- Physical defenses;
- Architectural changes;
- Trusted Computing