Education

ABCs of Data Encryption

Roger Cummings, Symantec

SNIA Legal Notice

- The material contained in this tutorial is copyrighted by the SNIA.
- Member companies and individuals may use this material in presentations and literature under the following conditions:
  - Any slide or slides used must be reproduced without modification
  - The SNIA must be acknowledged as source of any material used in the body of any document containing material from these presentations.
- This presentation is a project of the SNIA Education Committee.

Abstract

ABCs of Data Encryption

Public disclosures of data “indiscretions” have become regular enough and embarrassing enough that many organizations are exploring encryption options to simply stay out of the headlines. Those who have ventured into this space quickly realize that there is no “magic crypto fairy dust” that will make the problems go completely away. However, with careful planning and judicious use of the right technologies, organizations can eliminate many of their exposures.

This session focuses on the efforts required at the storage layer to create a successful encryption strategy. Major uses along with factors to consider are presented for protecting storage management, data in-flight, and data at-rest. The session provides expanded coverage on encrypting data at-rest, based on a step-by-step approach.

Notice

- The author is NOT an attorney, and nothing in this presentation is intended to be nor should be construed as legal advice or opinion. If you need legal advice or a legal opinion, please contact an attorney
- The information presented herein represents the author's personal opinion and current understanding of the issues involved. The author, Symantec, and SNIA do NOT assume any responsibility or liability for damages arising out of any reliance on or use of this information

Agenda

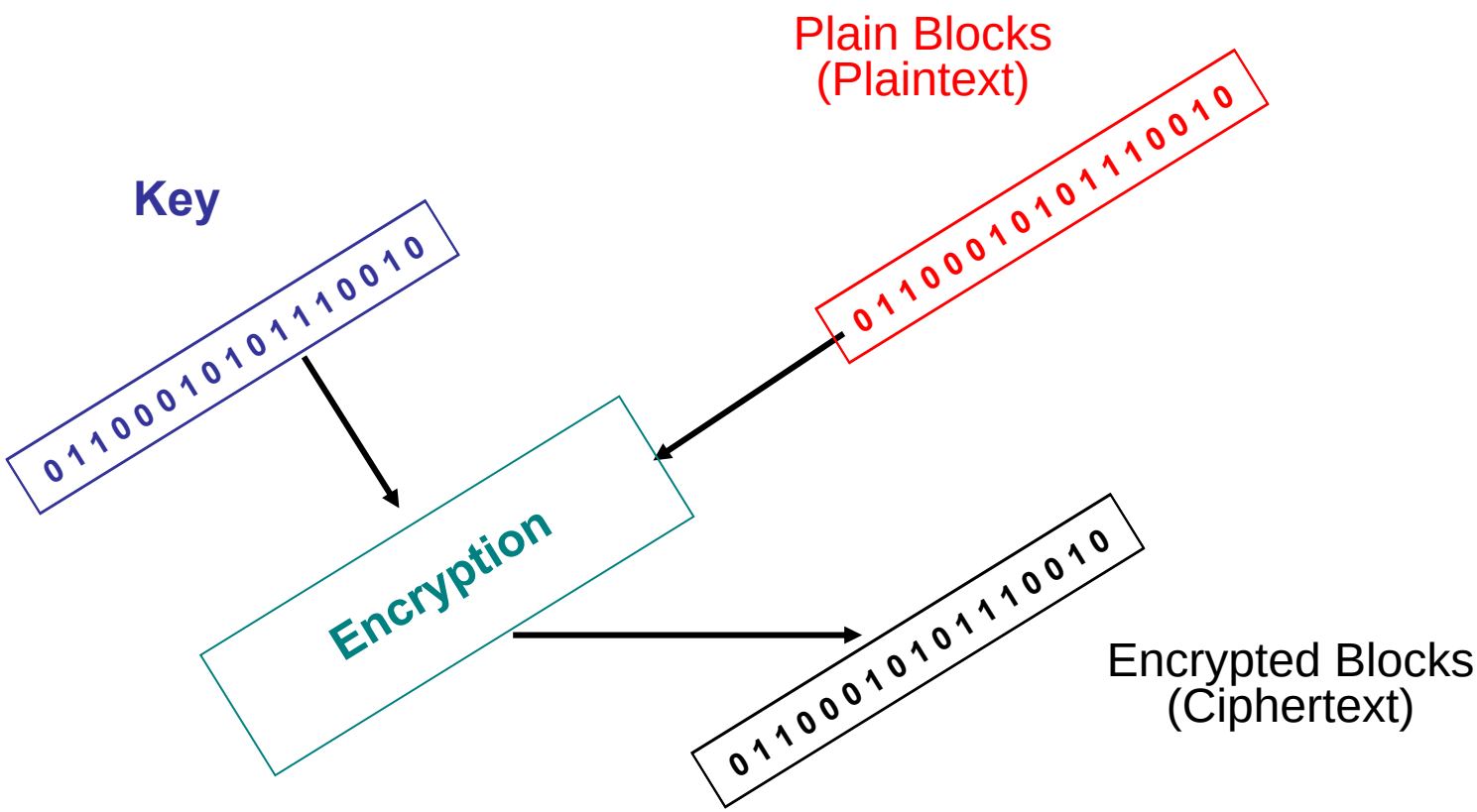
- Formal Definition of Important Terms
- BRIEF Encryption Background
  - More in backup
- Storage-related encryption in 2007
- The SNIA Nine Step Checklist
  - How to implement encryption in YOUR organization
- To get involved….
- Best Practices (under development)
- Sources of additional information

Important Terms

A Few Formal Definitions

- Plaintext – Original information (intelligible) that is used as input to an encryption algorithm (cipher).
- Ciphertext – The encrypted (unintelligible) output from an encryption algorithm.
- Encryption – The conversion of plaintext to encrypted text with the intent that it only be accessible to authorized users who have the appropriate decryption key.
- Cipher – A mathematical algorithm for performing encryption (and the reverse, decryption).
- Key – A piece of auxiliary information used by a cipher during the encryption operation.

Encryption Scheme Concept

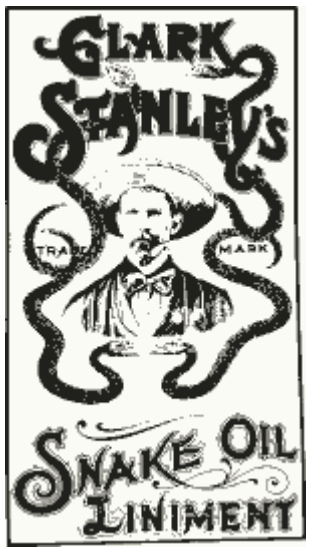


Encryption Background
(Warning – uses analogies)
(Any cryptographers present might want to tune out for a while)

Encryption is…..

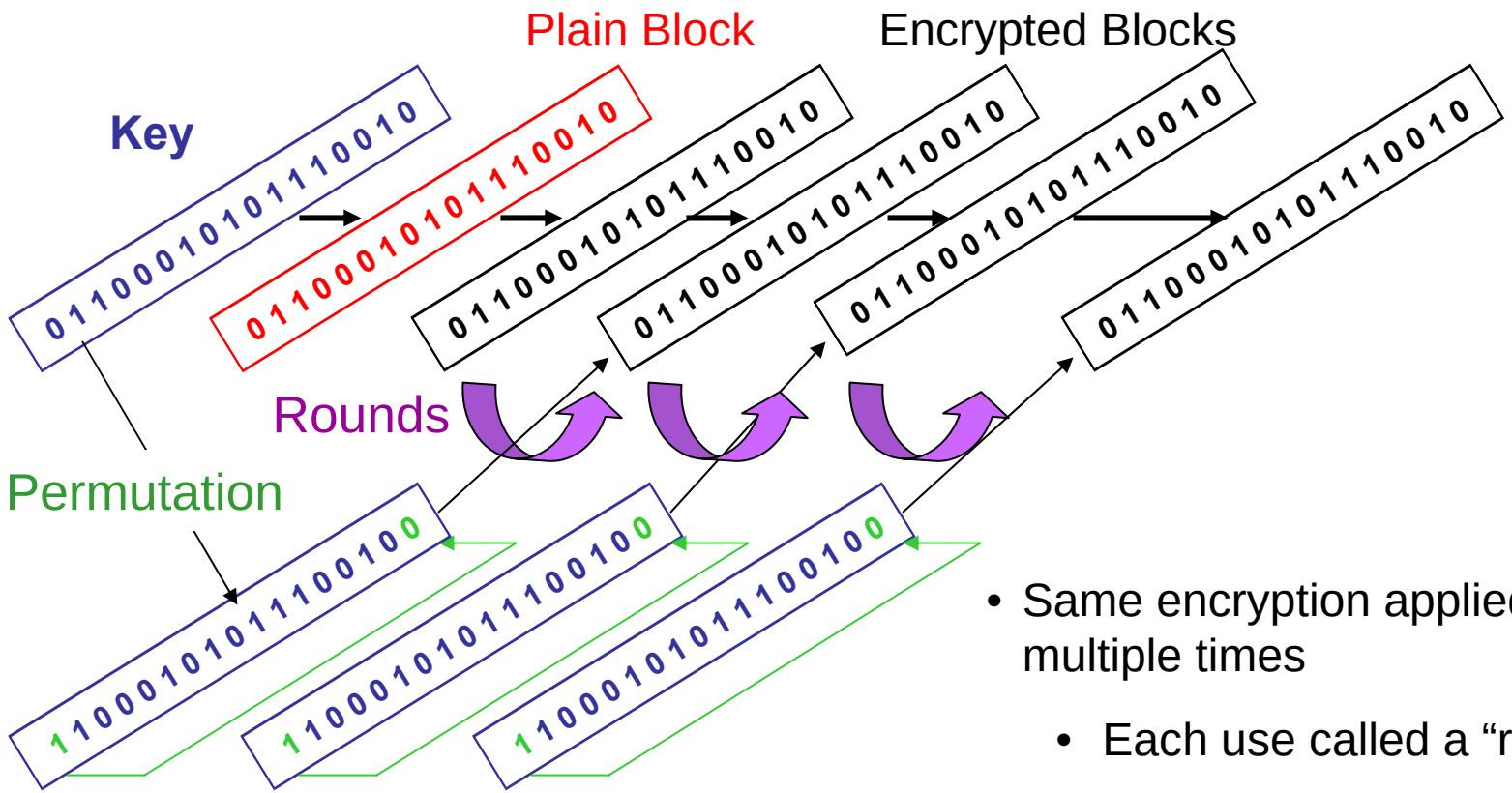
- Basically exactly what we knew as “secret writing” when we were kids
  - Remember the “magic decoder rings” given away in the comics?
    - `ABCDEFGHIJKLMNOPQRSTUVWXYZ`
    - `DEFGHIJKLMNOPQRSTUVWXYZABC`
  - Replace the letter in top line by the one underneath (substitution)
  - Reverse the order of the letters in the message (transposition)
- The above scheme is known as the Caesar Cipher
  - Reputed 1st used by Julius Caesar during the Gallic wars
- Arguably there have only been two major advances since the time of Caesar
- Cryptography (Greek “cryptos" (hidden) + "graphia" (writing))

Principles (or lack thereof)



- Kerckhoffs Principle (1883)
  - Only the key needs to be secret; there should be no secrecy in the algorithm
- Encryption is empirical science
  - We know a scheme is strong ONLY because it hasn't been broken yet…..
  - And have reasons for believing it's unlikely to be broken in a specific period using known or anticipated technology
- Use only published & proven schemes
  - And beware of snake oil!

First Advance - Mechanization



- Same encryption applied multiple times
  - Each use called a “round”
  - Cryptographic “mode” defines how output of one round becomes input of another

Second Advance - Asymmetry

- All previous encryption schemes were “symmetric”
  - Same key used to encrypt & decrypt
  - Thus key had to be distributed to both parties before communication could take place
- Major advance developed in the 1970s in area of key distribution
  - Based on “one way” mathematical functions
    - Not easy to determine the starting point from the result
  - Solved shared key “chicken & egg” problem
    - Now two parties that had never communicated before could do so securely
- First practical scheme was Diffie-Hellman Key Exchange
  - But it required a number of handshakes
    - Improved by RSA asymmetric cryptography

Analogy

- Chechen postal service said to be corrupt
  - Anything valuable sent in unlocked box is stolen
- How can Ramzan in Grozny send a valuable antique to Madina in Argun who he's never met
  1. He sends a box locked with his padlock
  2. She attaches a padlock of her own and sends it back
  3. He removes his padlock and sends the box back again
  4. She opens the box
- Could be a single step process IF Ramzan could get a "Madina lock" i.e. a lock that can be opened with a key that Madina already has, from his local post office
  - Important question - how & why should Ramzan trust the post office to give him the right thing?

Asymmetric Cryptography

- Two keys, one Public (open), one Private (secret)
- What one encrypts, the other can decrypt
  - And vice versa
- Cannot feasibly calculate one key given the other
- Also called Public Key Infrastructure (PKI) Cryptography
  - Because needs a TRUSTED infrastructure to distribute the public keys

Hashing

- Hashing does not encrypt data, but provides transformation used to verify data integrity
  - Hash algorithm digests data and represents its bits and bit patterns by fixed-size equivalent - a Hash Value
    - Size of the value is fixed by the algorithm (SHA-1 is 20 bytes)
    - Algorithm is non-reversible: cannot reproduce data from hash
    - Single bit change in data may change half of the bits in hash
- Does not require the use of keys
  - But there's a related construct called a Message Authentication Code (MAC) that uses a hash derived from both data and a secret key
    - HMAC is the best known today – see IETF RFC 2104 for details
- Hash mainly used to ensure data integrity and as "digital signature"

Summary

- The basics of cryptography have been known since the time of the Greeks & Romans
  - Same basic mechanisms have been used throughout that time
  - History gives us a good basis for understanding what's practical today
- Modern block ciphers use large alphabets and keys
  - Mechanization provides multiple substitutions and transpositions in series to add strength
- Asymmetric cryptography a truly revolutionary advance
  - Too compute-intensive for bulk data operations
  - But VERY useful for distributing keys

Some people change when they see the light, others when they feel the heat.

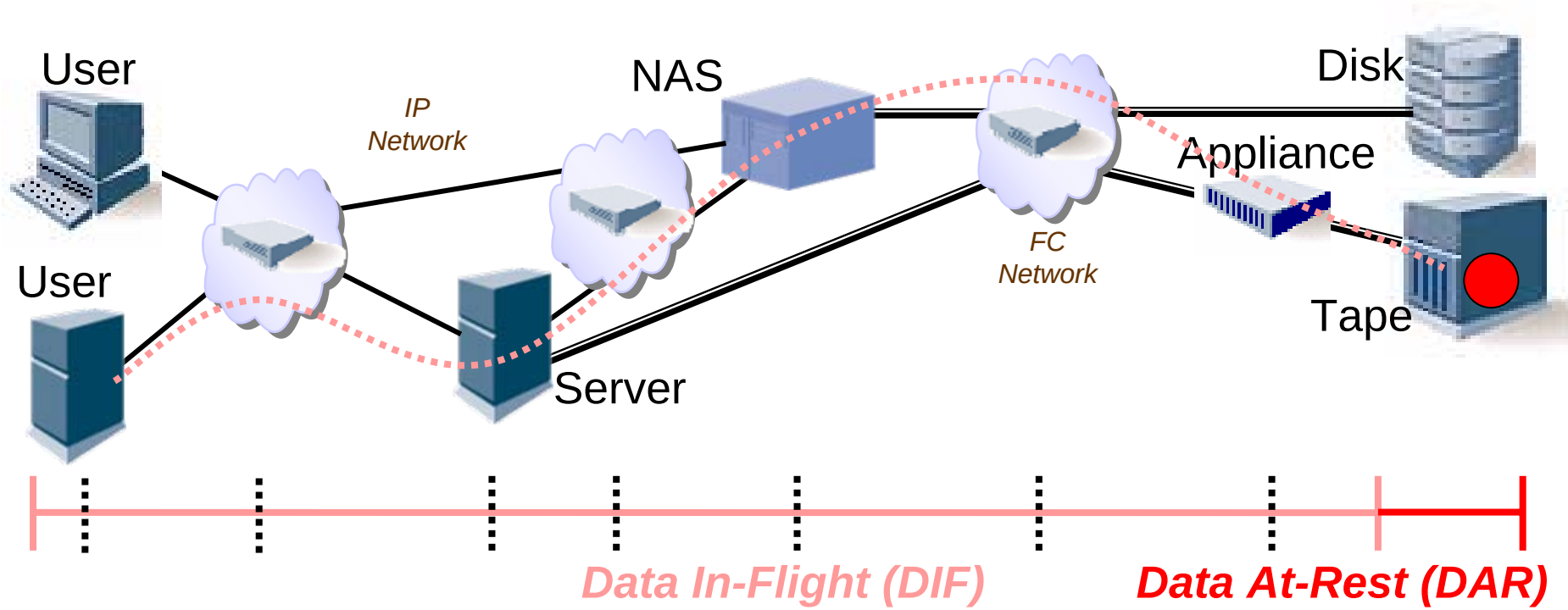
Caroline Schoeder

Storage-related Encryption

Lately, there have been…..

- A **lot** of storage security product announcements addressed at preventing repeats of past data "indiscretions"
  - Fueled by "lost tapes" & "lost laptop" scenarios
- A **lot** of confusion about data "in-flight" versus data "at-rest" security
- A **lot** of press about keys & related difficulties
  - Human involvement (e.g. policy creation, cross-group interaction) the source of much difficulty
- A **lot** of new readers of relevant published guidelines (e.g. NIST SP 800-57 Part 1)

In-Flight versus At-Rest



In-Flight:
- Two end points (communication)
- Interoperability – network layers
- Data is transitory (temporary)

At-Rest:
- Interoperability – media interchangeability
- Data is persistent on media

Yes, the term is a misnomer because media moves!

Data In-Flight

- Technology differs by “transport”
  - Block-level IP protocols
    - IPsec for iSCSI, iFCP, FCIP
  - Block-level, FC protocols
    - FC-SP, ESP_Header, CT protection
  - File-level, IP protocols
    - IPsec for NFS & SMB/CIFS
    - SSL/TLS for WebDAV
  - Management protocols
    - SSL/TLS or SSH for SMI-S, SNMPv3, web-based mgmt
- IPsec & TLS largely proven
  - Widely deployed for VPNs, less so for traffic inside the corporate firewall
- FC-SP starting to be deployed

Data At-Rest

- Encryption/decryption built into tape drives
  - Encryption AFTER compression (to keep usual ratio)
  - Key not stored on the media or retrievable from drive
    - Key-associated data to help in “found tape” case
  - Tape to tape copy without decryption being worked
- Disks that encrypt data before storing on media
  - Tied to attached system in some way
  - Can probably be reset by extraordinary means in field
    - But will certainly wipe all existing data
- Both of the above based on standard interfaces
  - Utilizing new features in SCSI and ATA interfaces and command sets

Data At-Rest (2)

- Security appliances
  - Most also compress data before encryption (to keep historical efficiencies)
  - Also include key management functions
- Applications with encryption features
  - Many have been around for quite a while!
- New OS offerings & encrypting file systems
- New NAS & CIFS products will emphasize security
  - New purpose-designed cryptographic schemes
  - New corporate alignments play a part!

Summary

- New storage products with data encryption are becoming available that address major users concerns
- The SNIA Nine-Step Checklist defines the tasks you need to complete to best utilize these products in your organization

The SNIA Nine Step Checklist

Introduction

- Step-by-step listing of tasks to be performed to effectively implement at-rest data encryption
  - Defines a process, not a single activity
  - Not all substeps will be needed in all cases, but they all merit consideration
- Annexes contain useful additional checklists related to security & encryption from:
  - Federal Financial Institutions Examination Council (FFIEC)
  - Information Systems Audit and Control Association (ISACA)
  - Payment Card Industry (PCI) Data Security Standard (DSS)

The Steps

1. Understand Confidentiality Drivers
2. Classify the data assets
3. Inventory data assets
4. Perform data flow analysis
5. Determine the points-of-encryption
6. Design the encryption solution
7. Begin data re-alignment
8. Implement Solution
9. Activate encryption

#1 Understand Confidentiality Drivers

- Identify regulatory obligations (Sarbanes-Oxley, HIPAA, PCI DSS, EU Data Privacy etc.)
- Identify legal obligations
  - Review recent audits & any legal interactions
- Talk with executive management about concerns
  - Real ones are the ones that get funded!
- Look @ IS/IT strategic plans

#2 Classify the data assets

- For the most part you cannot afford to encrypt everything
- Use coarse classifications to start
  - Mission critical
  - Most Sensitive
  - Regulated
  - Refine over time
- Determine confidentiality priorities & categories

Roger's Law of Security (2006 version)

- The Law says that as Security increases……
  - Cost **Increases**
  - Complexity **Increases**
  - Performance **Decreases**
  - Operational Efficiency **Decreases**
- But the MAGNITUDE of the changes varies a great deal by approach & situation

#3 Inventory data assets

- For each category determine
  - Equipments that transfer the data
  - Applications that process the data
  - Devices used to store the data
  - Networks used to transfer the data
    - Specifically those that leave the data center
  - Groups & people that own & are dependent on the data
- Perform risk analysis

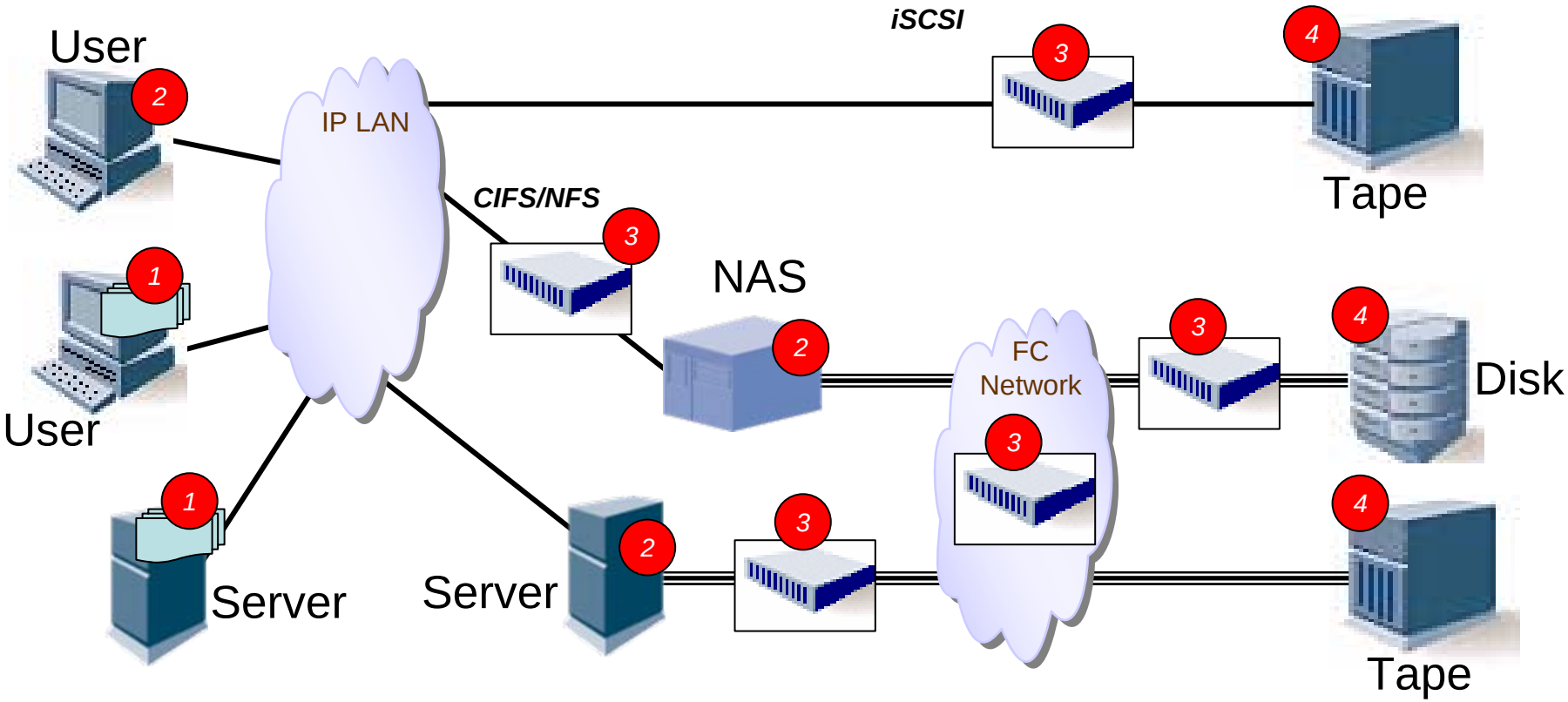
#4 Perform data flow analysis

- Look for temporary as well as permanent storage locations
- What about mobile devices?
  - How are laptops handled?
  - Blackberry? PDA?
- Don’t forget data protection schemes
  - Where’s that device mirrored or replicated?

#5 Choose points-of-encryption

- Security Perspective: Encrypt as close to source as possible.
- Points of Encryption:
  - **Application-level** – under the control of specific app or DB; finest granularity of control & max insight into data (type, users, sensitivity)
  - **Filesystem-level** – under the control of OS or OS-level app; control at file-level with insights into users
  - **Network-level** – under control of a network-based system
    - File-based (NAS) – control at share/filesystem-level (possibly file-level) with moderate insights into users
    - Block-based – control at logical volume level with limited or no insights in the "community of users"
  - **Device-level** – under the control of end-device; control at logical volume level with limited insights into "community of users"

Points of Encryption



#6 Design encryption solution

- Documentation is key here!
- Define a framework
  - Address key management structure – particularly where managed, how communicated, who's responsible
- Imagine having to demonstrate the protection to an auditor (or prove it to your legal department)
  - Do you collect the necessary information?
  - Can you demonstrate the chain of evidence?
- What's the impact on performance and/or operational effectiveness?
  - See "Law" in #2

#7 Begin data re-alignment

- Previous steps will probably require migration of data between devices and/or networks
  - Bandwidth & latency will change
    - Not everyone will be happy
  - May require infrastructure changes to address issues
    - If so do it now BEFORE going further
  - Don’t forget to change data protection schemes as well
    - Frequencies may change
    - New platforms may need to be utilized

#8 Implement Solution

- Determine approach to solution (outsourced, phased etc.)
- Create a rollback plan in parallel with determining the approach!
- Select technology & acquire components
- Deploy and integrate with key management
- Integrate with authentication, audit logging, directory services (access control)
  - Secure timestamp source very important

#9 Activate Encryption

- Activate encryption? NO, not quite yet!
- First get management signoff
  - Outside accreditation might be a good idea
- Complete final data realignment
- Might need to encrypt existing data in background first
- Run the system in “audit” mode for a while
  - Makes sure the right keys are available & logs working
- Only THEN turn encryption on for active data

Summary

- Several new secure storage product types have been announced or will be shortly
  - Based on industry standard definitions!
- Addressing major storage security concerns becomes truly feasible for the first time
- This tutorial has:
  - Defined the terminology
  - Introduced the underlying protocols & approaches
  - Laid out a process to follow when fielding the new storage security products

To Get Involved..

SNIA Security

- SNIA Security Technical Work Group (TWG)
  - Focus: Requirements, architectures, interfaces, practices, technology, educational materials, and terminology for storage networking.
  - http://www.snia.org/tech_activities/workgroups/security/
- Storage Security Industry Forum (SSIF)
  - Focus: Marketing collateral, educational materials, customer needs, whitepapers including Encryption of Data At-Rest (a Step-by-Step Checklist)
  - http://www.snia.org/ssif/documents

Draft BCPs - Encryption for Storage

- ENC01 – Protect Externalized Data (BCP#7)
  - ENC01.A Secure Sensitive Data on Removable Media
  - ENC01.B Secure Sensitive Data Transferred Between Data Centers
  - ENC01.C Secure Sensitive Data in 3rd-party Data Centers
- ENC02 – Pedigree of Encryption
  - ENC02.A Encryption Algorithms
  - ENC02.B Symmetric Encryption Modes
  - ENC02.C Strength of Encryption
- ENC03 – Risk Assessment in Use of Encryption
  - ENC03.A Identify and Classify Sensitive Data
  - ENC03.B Analyze Risks and Protection Options
  - ENC03.C Mitigate Risks with Encryption

Draft BCPs - Key Management Services

- KMS01 – Key Management Principles
  - KMS01.A Observe Important Properties of Keys
  - KMS01.B Use and Implement KM Safely
- KMS02 – Key Management Functions
  - KMS02.A Establish Keys Securely
  - KMS02.B Ensure Proper Operational Use
  - KMS02.C Disassociation & Disposition
- KMS03 – Key Management Issues
  - KMS03.A Comply with Import/Export Controls
  - KMS03.B Plan for Problems

Q&A / Feedback

- Please send any questions or comments on this presentation to SNIA: tracksecurity@snia.org

Many thanks to the following individuals for their contributions to this tutorial.

SNIA Education Committee

Eric A. Hibbard, CISSP
Andrew Nielsen, CISSP
Sean Gettman
LeRoy Budnik
Phil Huml
Curt Kolovson

SNIA Security TWG

Jim Norton
Richard Austin, CISSP
Larry Hofer CISSP
Robert Lockhart
David Black
Roger Cummings

SNIA SSIF

For More Information

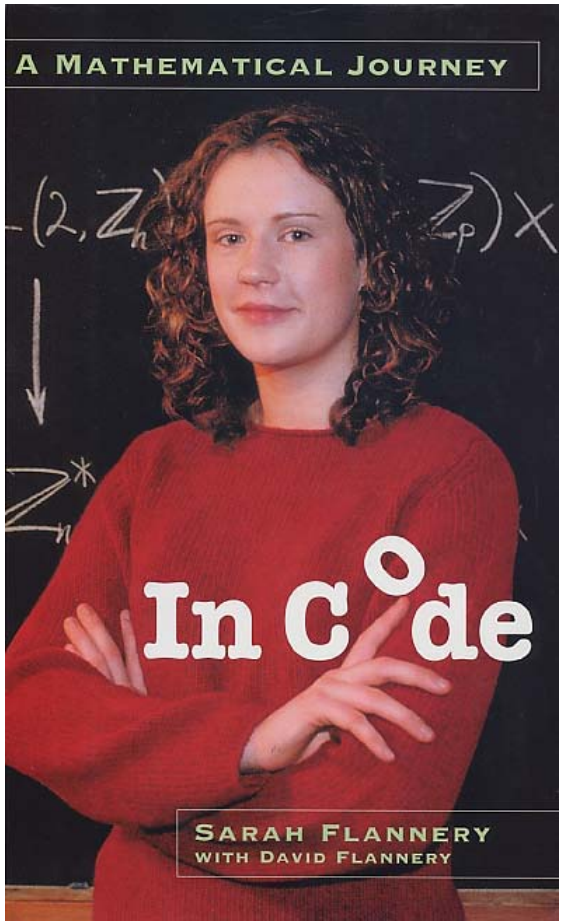
Relevant Storage Security Standards Bodies

- ISO/IEC JTC1 SC27 (www.din.de/ni/sc27) – IT Security Techniques
  - US group is ANSI/INCITS CS1 (www.cs1.incits.org)
- NIST/CSD Computer Security Resource Center (csrc.nist.gov) – Security standards for US Government
- IEEE/P1619 (siswg.net) – Security in Storage Working Group
- ANSI/INCITS T10 (www.t10.org) – SCSI security, tape drive encryption control etc.
- ANSI/INCITS T11 (www.t11.org) – Fibre Channel security (FC-SP)
- IETF (www.ietf.org) – IP security (IPsec), Transport Layer Security (TLS)

Web Sources of Information

- The CERT® Coordination Center, http://www.cert.org
- The SANS (SysAdmin, Audit, Network, Security) Institute, http://www.sans.org
- The Center for Internet Security (CIS), http://www.cisecurity.org
- Information Systems Audit and Control Association (ISACA) – *IS Standards, Guidelines, and Procedures for Auditing and Control Professionals*, http://www.isaca.org/standards/
- Information Security Forum (ISF) – The Standard of Good Practice for Information Security, http://www.isfsecuritystandard.com/

Maths Reference

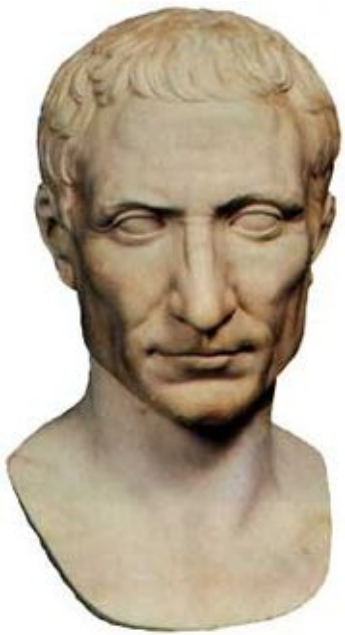


ISBN 1-565123-77-8

- The story of an Irish schoolgirl's science project
  - That won Ireland's 1999 Young Scientist of the Year award!
  - An asymmetric cryptography scheme based on matrix multiplication
    - Turned out to be insecure, but so what!
- A simple introduction to the maths underlying cryptography
  - And a great story in general

Backup Slides

Caesar Cipher



- A Monoalphabetic Substitution Cipher
  - Only 25 keys possible
  - Can extend this by jumbling letters rather than shifting
  - Use keywords to define 26 bit key more easily e.g. keyword of “secret writing” give following substitution alphabet (unique letters from keyword followed by others in order)
    - `ABCDEFGHIJKLMNOPQRSTUVWXYZ`
    - `SECRTWINGHJKLMNOPQUVXYZBDF`
  - Total of 26! (4 x 10^{26} keys) for a jumble

Key Space

- Number of possible keys a function of alphabet
- How to increase the number of possibilities??
  - Encoding 2 letters at a time gives 676 values
  - Encoding 3 letters at a time gives 17576 values
  - Encoding 4 letters at a time gives 456976 values etc.
- Modern block encryption schemes use 64 -256 bit blocks
  - Gives a very large alphabet!!
- However the basis of the schemes is still substitution & transposition (permutation)
  - Now commonly called S-Box & P-Box for some reason
    - http://en.wikipedia.org/wiki/S-box
    - http://en.wikipedia.org/wiki/permutation

Stupid Exhaustive Key Search

Key Size (bits)	Time (1μs/test)	Time (1μs/10^6tes
32	35.8 mins	2.15 mse
40	64.8 days	550 mse
56	1140 years	10.0 hou
64	~500,000 years	107 days
128	5 x 10^{24} years	5 x 10^{18} years

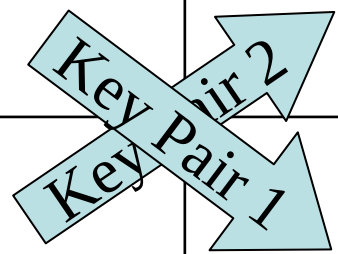
Many things can reduce all these times substantially e.g. smart use of mathematics, weak keys, known plaintext & algorithm quirks

For comparison, the remaining lifetime of the Earth is currently estimated at 4 x 10^9 years

See http://www.historyoftheuniverse.com

Public Key Usage Explained

	Encrypt	Decrypt
Public Key Operation (very **fast**)	Encryption	Verifying
Private Key Operation (very very **slow**)	Signing	Decryption



Source: Bob Thibadeau, Seagate

RSA Asymmetric Cryptography

- Rivest, Shamir, Adelman, work done @ MIT, late '70s
- Setup
  - P and Q are large prime numbers that remain secret!
  - N = PQ, M = (P-1)(Q-1)
  - E is a (normally small) choice which is less than & relatively prime to M
  - D = multiplicative inverse of E modulo M
  - Publish N and E as Public Key, keep D private
- Encrypting: Ciphertext = $(\text{Plaintext\#})^{E}$ modulo N
- Decrypting: Plaintext = $(\text{Ciphertext\#})^{D}$ modulo N
- Exponentiation using large powers is compute intensive
- Factorization of large numbers is VERY difficult
- Plaintext# is a number whose base 256 representation is the plaintext message block being encrypted

RSA Worked Example

- Use simplified 3 letter alphabet a=0, b=1, c=2
- So the string “abc” becomes $(0x3^2+1x3+2) = 5$
- Set P=11 & Q=3 (would be 100+ digits in practice)
- N=11x3=33, M=10x2=20
- Choose E as 3 as being relatively prime to an M of 20
- Find D such that 3xD mod 20=1 i.e. D=7
- Encryption 5^3 mod 33 = 125 mod 33 = 26
  - = $(2x3^2+2x3+2)$ becomes “ccc”
- Decryption 26^7 mod 33 = $26^{(3+3+1)}$ mod 33
  - = $((26^3 \bmod 33)^2 . 26) \bmod 33$
  - = $((26^2 \bmod 33). 26) \bmod 33)^2 . 26) \bmod 33$
  - = $(16.26 \bmod 33)^2 . 26) \bmod 33$
  - = $20^2.26 \bmod 33 = 10400 \bmod 33 = 500 \bmod 33 = 5$ becomes “abc”

Theorem:

ab mod c = ((a mod c) (b mod c)) mod c

Two Types Combined