I. Introduction
Who is StrongAuth?

• Cupertino CA-based private company
• Founded in 2001
• Focused on Architecture, Design, Development & Support of:
  − Enterprise Key Management
    • Public Key Infrastructure (PKI)
    • Symmetric Key Management System (SKMS)
• Customers in many sectors
  − Finance, Pharmaceutical, Medical Devices, e-Commerce, Entertainment, Retail, BPO Services, Manufacturing
Why bother listening to me?

• 30+ years of work-experience
  − 6 years on the Business side
  − 24+ in Information Technology
  • 10+ in Cryptographic Key Management

• Designer, lead-developer of StrongKey – the industry's first, open-source, Symmetric Key Management System (2006)

• Designer, lead-developer of the StrongKey Lite Encryption System – the industry's lowest cost encryption & KM appliance (2010)
II. Some Definitions
• Encryption
  - A reversible cryptographic operation that transforms meaningful “plaintext” to illegible “ciphertext”

• Tokenization
  - A reversible operation that substitutes meaningful “plaintext” to meaningless “plaintext”

• Hashing
  - An irreversible cryptographic operation that transforms meaningful “plaintext” to an illegible message-digest (hash)

• Key Management
  - The life-cycle operations associated with the secure creation, use, management, distribution and destruction of cryptographic keys
Symmetric Encryption

• The process of transforming plaintext to ciphertext, and vice-versa, using the same encryption/decryption key

You must be the change you want to see in the world.

M.K. Gandhi

Encryption + eaf0f527e6b4cfe52e8 + Decryption

50333904b5042a5c0e88a9b74211b67ee3aabb918744405799b30d611010824e6c0fd81faa34e245
Symmetric Encryption

- **Shared key** for encryption and decryption
- Faster
- Unlimited size for plaintext
  - Typically used to encrypt bulk data
- **Data Encryption Standard (DES) – 56-bit**
- **Triple-Data Encryption Standard (3DES)**
  - 112 and 168-bit
- **Advanced Encryption System (AES)**
  - 128, 192 and 256-bit
Asymmetric Encryption

- The process of transforming *plaintext* to *ciphertext*, and vice-versa, using *two different* keys

You must be the change you want to see in the world.  
*M.K. Gandhi*
Asymmetric Encryption

• **Different** keys for encryption & decryption
• Slower
• Limited size for plaintext
  - Less than the size of the key
  - Used to encrypt symmetric keys & hashes

• Rivest-Shamir-Adelman (RSA)
  - 512 to 8192-bits
  - 2048-bits recommended for 2010 deployments
Message Digest (Hash)

• The object created by the process of transforming data to a **fixed-size** cryptographic value using a **one-way** transformation process.
Message Digest (Hash)

• No key is involved – just an algorithm
• Unlimited size data
• Typically used to verify the integrity of a file
• Message Digest 5 (MD5) – Broken!!
  – 128-bit fixed size
• Secure Hashing Algorithm – (SHA)
  – SHA1: 160-bit (Avoid, if possible)
  – SHA-256, SHA-384 and SHA-512
Tokenization

- The process of **substituting** a like-value for plaintext **without** the use of cryptography

1234 5678 9012 3456

9999 0000 0000 5678

123-45-6789

800-00-0123

123456789 98765432

100000000 00001234
III. Cryptography Pitfalls
Cryptography pitfalls-1

- Storing symmetric key in a file, registry-entry, database record – *somewhere on the system*
- Encrypting symmetric key with public key, but storing private key in a file
- Using Password-Based-Encryption (PBE), but storing the password in a file
- Compiling symmetric key into the program
- Encrypting symmetric key with another symmetric key
- Backing up the key with the ciphertext
• Using a single key to encrypt all data
• Not verifying the integrity of decrypted data
• Not thinking through key-rotation issues
  – Single rotation per year
  – Rotating DEK-ciphertext - not data-ciphertext
• Not thinking through split-key knowledge issues
• Not planning for rapid changes in cryptography
• Encrypting at the wrong layer of the stack
Real-world analogy

Precious cargo is protected all the time!
Real-world analogy
• Encrypting at the wrong layer of the stack
The right way

Application
Network
Web Server
Application Server
JDBC/ODBC
Database
Operating System
Hard Disk Drive

Irrelevant
IV. Solution
So, what do you do?

- Reduce the exposure of sensitive data
- Abstract cryptography **out** of the application
- Use a cryptographic hardware module as a back-stop
- Use specialized solutions rather than “home-brewed” encryption
- Follow NIST guidelines for algorithms, key-sizes
Reduce the exposure - 1

System-1
Sensitive Data

System-2
Sensitive Data

System-3
Sensitive Data

System-4
Sensitive Data

System-5
Sensitive Data
Reduce the exposure -2

System-1
Sensitive Data

System-2
Sensitive Data

System-3
Sensitive Data

System-4
Sensitive Data

System-5
Sensitive Data
Reduce the exposure - 3
Use cryptographic hardware

• **Trusted Platform Module**
  - CC EAL4+ certified
  - RSA 2048-bit keys that never leave the TPM
  - Embedded on computer motherboards

• **Hardware Security Module**
  - FIPS 140-2 certified
  - RSA and Suite-B algorithms
  - Erases on-board cryptographic material when stolen
Use specialized solutions
• Triple-DES (112- or 168-bits) symmetric keys
• AES (128-, 192- or 256-bits) symmetric keys
• RSA (2048-bits or greater) asymmetric keys
• SHA-256, SHA-384 or SHA-512 for message-digests
• FIPS 140-2 certified cryptographic hardware modules
• Common Criteria EAL certified cryptographic hardware modules
Summary

• Cryptography has always been complex, but is getting increasingly so:
  – Attackers are knowledgable and using crypto
  – Crypto-hardware is becoming ubiquitous
  – Growing number of crypto forums and standards
  – State laws are referencing PCI-DSS or crypto directly
    • Massachusetts, Minnesota, Nevada, Washington

• Education and a long-term strategy is key to preventing crypto-chaos
• Thank You
• Questions?
• Contact Information:
  − Arshad Noor
  − arshad.noor@strongauth.com
  − (408) 331-2001 Direct
  − (408) 515-8557 Mobile
  − www.strongauth.com