Software Craftsmanship
McHenry County
November 14, 2012

The

Encryption Demolition

Michael Buselli

Takeaways

- Don't invent/implement your own cryptosystem.
- Encryption in motion: SSL/TLS, SSH, IPSec.
- Encryption at rest: PGP/GnuPG.
- Do not use AES, RSA, etc. algorithms directly.
- Avoid and delegate using crypto if possible.
- Exceptions do exist, but hire a crypto expert before you pass Go.
- Popular programs/libraries often get it wrong.

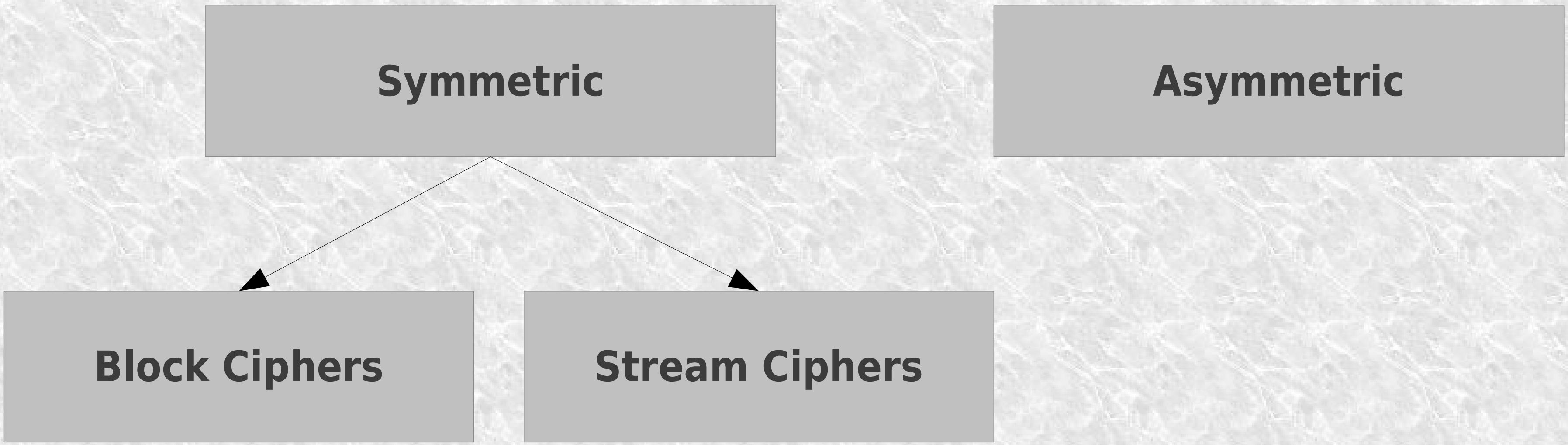
Why not do-it-yourself?

- Security is only as strong as weakest link.
- Bad algorithm?
- Bad implementation?
- Guessable or impressionable key generation?
- Appropriate symmetric block encryption mode?
- Tamper proof?
- Replay protection?

CIA: Why we do Encryption

- Confidentiality
- Integrity
- Authentication

Encryption Algorithms



Kerckhoffs's Principle

- "A cryptosystem should be secure even if everything about the system, except the key, is public knowledge." –Auguste Kerckhoffs
- Also known as avoiding "security through obscurity."
  - http://en.wikipedia.org/wiki/Kerckhoffs%27s_principle

Symmetric Block Ciphers

- Private key ciphers that encrypt by block.
- Examples: AES, 3-DES, Blowfish, Serpent
- Some modes allow use as a stream cipher.

Symmetric Stream Ciphers

- Private key ciphers that encrypt bit by bit.
- Examples: RC4, Salsa20
- Usually faster than block ciphers.
- No need for padding.

Asymmetric Ciphers

“Public Key Ciphers”

- Different encryption and decryption keys.
- Examples: RSA, DSA, ElGamal, Elliptic Curve Algorithms
- Usually much slower than symmetric ciphers.
- Typical key size 2048 and up.

ECB
Electronic Code Book

- Encrypts blocks one-to-one with ciphertext.
- “Hello to all... ”
  - → 51b8..35
- “Hello to all... Hello to all... ”
  - → 51b8..35 51b8..35

ECB

Electronic Code Book

CBC
Cipher Block Chaining

- Starts with an initialization vector XORed to first plaintext block.
- Each subsequent plaintext block is XORed to previous ciphertext.
  - encrypt(plaintext1 XOR init_vector, key) → ciphertext1
  - encrypt(plaintext2 XOR ciphertext1, key) → ciphertext2
  - encrypt(plaintext3 XOR ciphertext2, key) → ciphertext3
  - decrypt(ciphertext1, key) XOR init_vector → plaintext1
  - decrypt(ciphertext2, key) XOR ciphertext1 → plaintext2
  - decrypt(ciphertext3, key) XOR ciphertext2 → plaintext3

CBC
Cipher Block Chaining

- “Hello to all... Hello to all... ”
  - → 41c9...c9 1577...f8 610e...4c
- First block in ciphertext is a randomly generated initialization vector.

CFB and OFB

Cipher Feedback and Output Feedback

- Behave like a stream cipher.
  - keystream1 = encrypt(init_vector, key)
  - keystream1 XOR plaintext1 → ciphertext1
  - keystream2 = encrypt(**ciphertext1/keystream1**, key)
  - keystream_n = encrypt(**$\mathbf{ciphertext}_{\mathbf{n-1}}$/$\mathbf{keystream}_{\mathbf{n-1}}$**, key)
  - keystream_n XOR plaintext_n → ciphertext_n

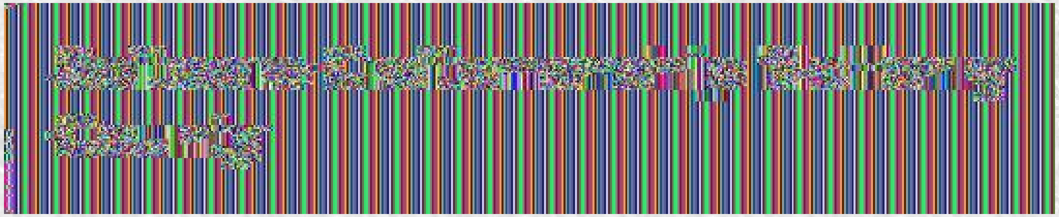
CTR
Counter Mode

- Generates keystream from sequentially increasing nonce.
  - keystream1 = encrypt(**nonce**, key)
  - keystream1 XOR plaintext1 → ciphertext1
  - keystream2 = encrypt(**nonce + 1**, key)
  - $keystream_n$ = encrypt(**nonce + (n - 1)**, key)
  - $keystream_n$ XOR $plaintext_n$ → $ciphertext_n$

Entropy

- Measure of level of uncertainty in data.
- Ciphertext should have high entropy.

~1.1 bit per byte



~5.4 bits per byte

~8.0 bits per byte

Password Selection

Bonus Slide!

- Passwords should have 40+ bits of entropy.
- English text has about 1 bit of entropy per letter.
- Completely random printable ASCII characters have about 6.5 bits of entropy per character.
- Randomly generated words have about 12 bits of entropy per word (depends on dictionary size).

Hash Functions

- Collision Resistance
- Examples: MD and SHA family
- Many use a symmetric block cipher in core.
- Caution: be wary of length extension attacks.

Password Hashes

- Tunable Slowness
- Examples: bcrypt, PBKDF2, scrypt

MAC

Message Authentication Code

- Provides message integrity.
- Examples: CBC-MAC, HMAC
- When using, encrypt then MAC ciphertext.
- Use different keys for encryption and MACing.

Authenticated Encryption

- Combines authentication and integrity in one mode.
- Examples:
  - GCM: Galois Counter Mode
  - OCB: Offset Codebook Mode (patented)
  - EAX: not a fancy acronym
  - CCM: Counter Mode with CBC-MAC

Punishing Bad Crypto

- Frequency Analysis
  - “MCHENRY COUNTY” → “ZPURAEL PBHAGL”
- Key Reuse
  - ciphertext1 = data1 XOR keystream
  - ciphertext2 = data2 XOR keystream
  - ciphertext1 XOR ciphertext2 = data1 XOR data2
- Padding Oracles

Coursera Crypto Class for a deep dive:
https://www.coursera.org/course/crypto

Questions?

Labs

Twitter: @cosine
Currently Ignored Blog: http://cosine.org/

https://github.com/cosine/Presentation-EncryptionDemolition.git