Symmetric Key Cryptography

Lecture 8

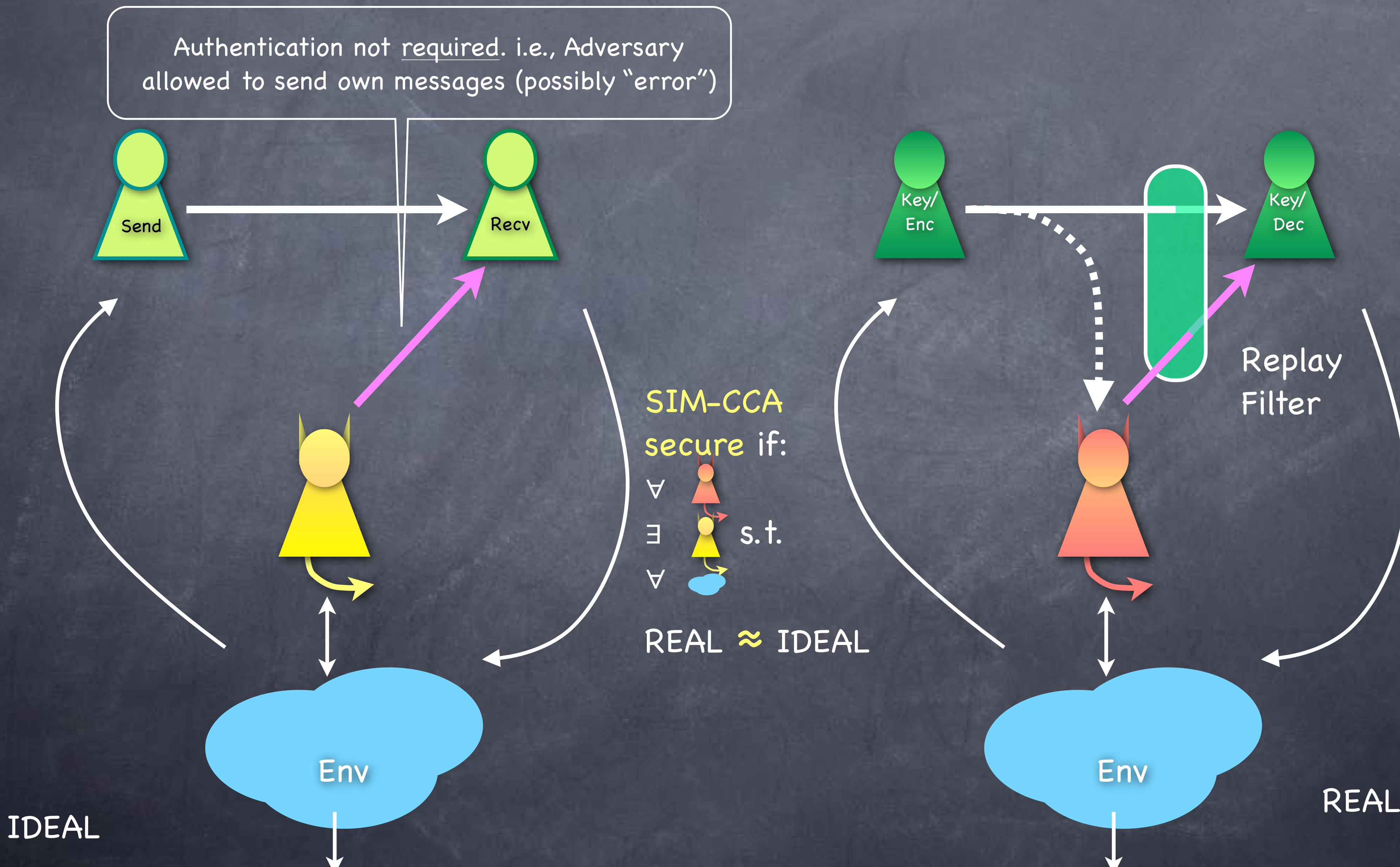
MAC

Summary

RECALL

Symmetric-Key Encryption

SIM-CCA Security



RECALL

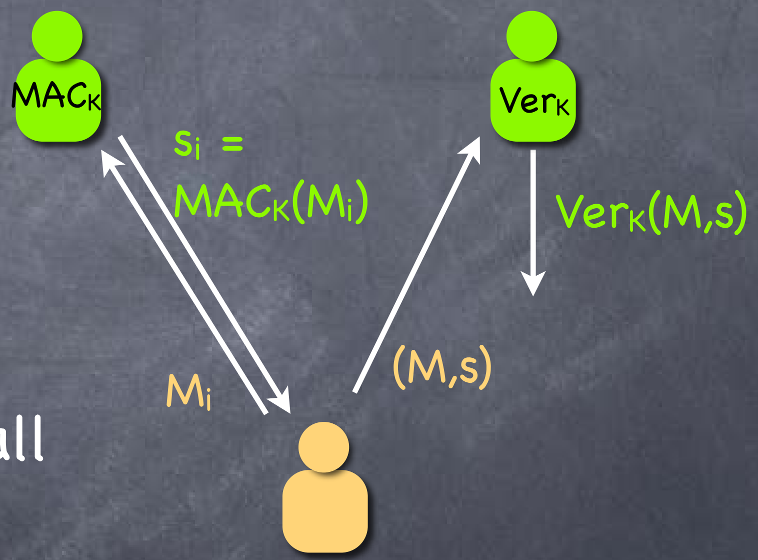
Encryption & Authentication

- CPA secure encryption: Block-cipher/CTR mode construction
- MAC: from a PRF or Block-Cipher
- CCA secure encryption: From CPA secure encryption and MAC. Encrypt-then-MAC. (Gives authentication also.)
- **SKE can be entirely based on Block-Ciphers**
  - A tool that can make things faster: Hash functions (later)

RECALL

Message Authentication Codes

- A single short key shared by Alice and Bob
  - Can sign any (polynomial) number of messages
- A triple (KeyGen, MAC, Verify)
- Correctness: For all K from KeyGen, and all messages M, $Verify_K(M,MAC_K(M))=1$
- Security: probability that an adversary can produce (M,s) s.t. $Verify_K(M,s)=1$ is negligible unless Alice produced an output $s=MAC_K(M)$



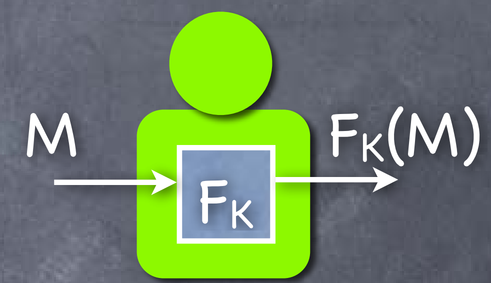
Advantage $= Pr[\ Ver_K(M,s)=1$ and $(M,s) \notin \{(M_i,s_i)\}\ ]$

RECALL

MAC from PRF

When Each Message is a Single Block

- PRF <u>is</u> a MAC!
  - $MAC_K(M) := F_K(M)$ where F is a PRF
  - $Ver_K(M,S) := 1$ iff $S=F_K(M)$
  - Output length of F_K should be big enough



- If an adversary forges MAC with probability ε_{MAC}, then can break PRF with advantage $O(\varepsilon_{MAC} - 2^{-m(k)})$ (m(k) being the output length of the PRF) [How?]
  - If random function R used as MAC, then probability of forgery, $\varepsilon_{MAC^*} = 2^{-m(k)}$

Recall: Advantage in breaking a PRF F = diff in prob test has of outputting 1, when given F vs. truly random R

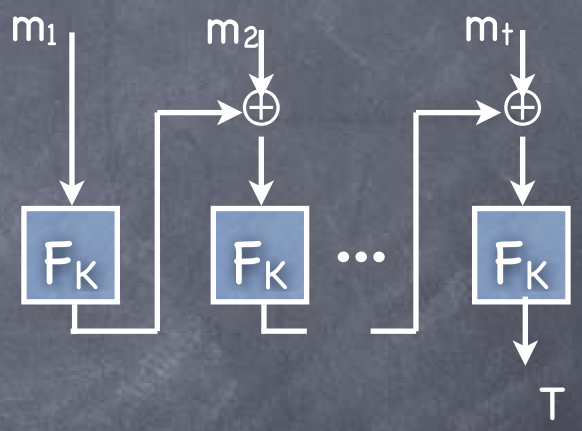
RECALL

MAC for Multiple-Block Messages

- A simple solution: "tie the blocks together"
  - Add to each block a random string r (same r for all blocks), total number of blocks, and a sequence number
    - $B_i = (r, t, i, M_i)$
    - $MAC(M) = (r, (MAC(B_i))_{i=1..t})$
    - r prevents mixing blocks from two messages, t prevents dropping blocks and i prevents rearranging
- Inefficient! Tag length increases with message length

CBC-MAC

- PRF domain extension: Chaining the blocks
  - cf. CBC mode for encryption (which is not a MAC!)



- t-block messages, a single block tag
- Can be shown to be secure
  - If restricted to t-block messages (i.e., same length)
  - Else **length-extension attacks** possible (by extending a previously signed message)

Patching CBC-MAC

- Patching CBC MAC to handle message of any (polynomial) length but still producing a single block tag (secure if block-cipher is):
  - Derive K as $F_{K'}(t)$, where t is the number of blocks
  - Use first block to specify number of blocks
    - Important that first block is used: if last block, message extension attacks still possible
  - EMAC: Output not the last tag T, but $F_{K'}(T)$, where K' is an independent key (after padding the message to an integral number of blocks). No need to know message length a priori.
  - CMAC: XOR last message block with a key (derived from the original key using the block-cipher). ← NIST Recommendation. 2005
- Later: Hash-based HMAC used in TLS and IPSec ← IETF Standard. 1997

Authenticated Encryption

- Doing encryption + authentication better
  - Generic composition: encrypt, then MAC
  - Needs two keys and two passes

> MAC-then-encrypt is not necessarily CCA-secure

- AE aims to do this more efficiently
  - Several constructions based on block-ciphers (modes of operation) provably secure modeling block-cipher as PRP
    - One pass: IAPM, OCB, ... [patented]
    - Two pass: CCM, GCM, SIV, ... [included in NIST standards]
  - AE with Associated Data: Allows unencrypted (but authenticated) parts of the plaintext, for headers etc.

SKE in Practice

Stream Ciphers

> Also used to denote the random nonce chosen for encryption using a block-cipher

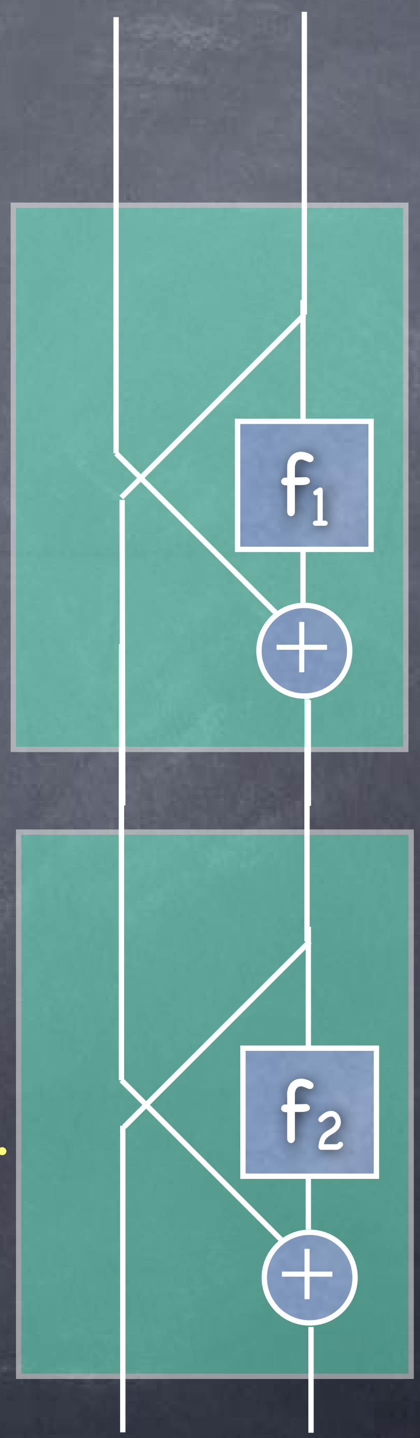
- A key should be used for only a single stream
- RC4, eSTREAM portfolio, ...
- In practice, stream ciphers take a key and an "IV" (initialization vector) as inputs
  - Heuristic goal: behave somewhat like a PRF (instead of a PRG) so that it can be used for multi-message encryption
  - But often breaks if used this way
- NIST Standard: For multi-message encryption, use a block-cipher in CTR mode

Block Ciphers

- DES, 3DES, Blowfish, AES, ...
  - Heuristic constructions
  - Permutations that can be inverted with the key
  - Speed (hardware/software) is of the essence
  - But should withstand known attacks
    - As a PRP (or at least, against key recovery)

Feistel Network

- Building a permutation from a (block) function
  - Let $f: \{0,1\}^m \rightarrow \{0,1\}^m$ be an arbitrary function
  - $F_f: \{0,1\}^{2m} \rightarrow \{0,1\}^{2m}$ defined as $F_f(x,y) = (\ y,\ x \oplus f(y)\ )$
    - F_f is a permutation (Why?)
      - Can invert (How?)
  - Given functions $f_1,\ldots,f_t$ can build a t-layer Feistel network $F_{f1\ldots ft}$
    - Still a permutation from $\{0,1\}^{2m}$ to $\{0,1\}^{2m}$
- **Luby-Rackoff**: A 3-layer Feistel network with PRFs (with independent seeds) as round functions is a PRP. A 4-layer Feistel of PRFs gives a strong PRP.
  - Fewer layers do not suffice! [Exercise]



DES Block Cipher

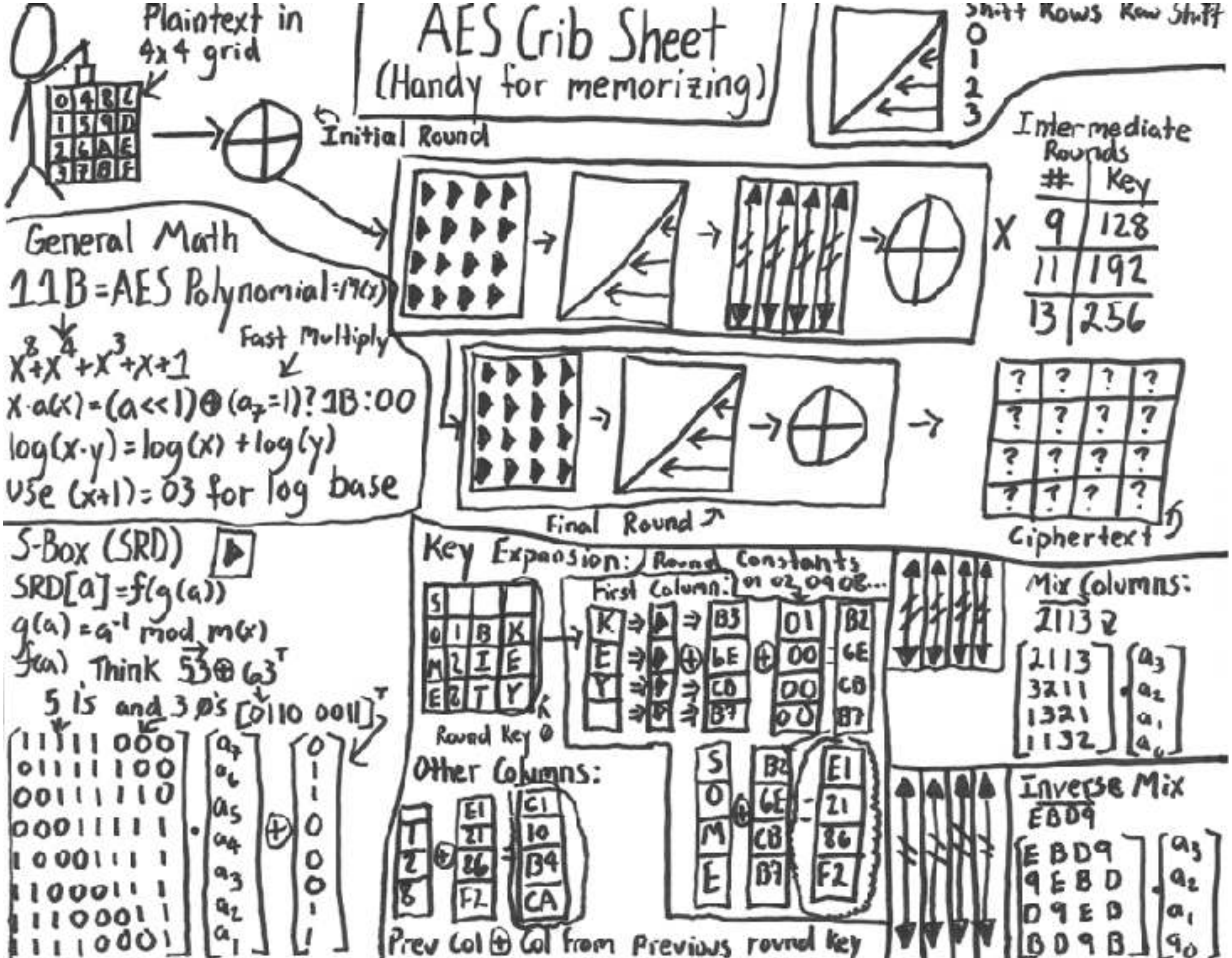
NIST Standard. 1976

- Data Encryption Standard (DES), Triple-DES, DES-X
- DES uses a 16-layer Feistel network (and a few other steps)
  - The round functions are not PRFs, but ad hoc
    - "Confuse and diffuse"
  - Defined for fixed key/block lengths (56 bits and 64 bits); key is used to generate subkeys for round functions
- DES's key length too short
  - Can now mount brute force key-recovery attacks (e.g. using $10K hardware, running for under a week, in 2006; now, in under a day)
- DES-X: extra keys to pad input and output
- Triple DES: 3 successive applications of DES (or DES^{-1}) with 3 keys

AES Block Cipher

NIST Standard. 2001

- Advanced Encryption Standard (AES)
  - AES-128, AES-192, AES-256 (3 key sizes; block size = 128 bits)
  - Very efficient in software implementations (unlike DES)
  - Uses "Substitute-and-Permute" instead of Feistel networks
  - Has some algebraic structure
    - Operations in a vector space over the field $GF(2^8)$
    - The algebraic structure may lead to "attacks"? Not yet.
  - Some implementations may lead to side-channel attacks (e.g. cache-timing attacks)
  - Widely considered secure, but no "simple" hardness assumption known to imply any sort of security for AES



By Jeff Moser (http://www.moserware.com/2009/09/stick-figure-guide-to-advanced.html)

Cryptanalysis

- Attacking stream ciphers and block ciphers
  - Typically for key recovery
- Brute force cryptanalysis, using specialized hardware
  - e.g. Attack on DES in 1998
- Several other analytical techniques to speed up attacks
  - Sometimes "theoretical": on weakened ("reduced round") constructions, showing improvement over brute-force attack
  - Meet-in-the-middle, linear cryptanalysis, differential cryptanalysis, impossible differential cryptanalysis, boomerang attack, integral cryptanalysis, cube attack, ...

SKE today

- SKE in IPsec, TLS etc. mainly based on AES block-ciphers
  - AES-128, AES-192, AES-256
- A recommended choice: AES Counter-mode + CMAC (or HMAC), encrypt-then-MAC.
  - Gives CCA security, and provides authentication
  - (Standards don't all follow this choice, but still secure)
- Older components/modes still in use
  - Supported by many standards for legacy purposes
  - In many applications (sometimes with modifications)
    - e.g. RC4 in BitTorrent, (older) Skype, PDF