601.445/601.645
Practical Cryptographic Systems

Symmetric Cryptography (cont'd)

Instructor: Alishah Chator

Housekeeping

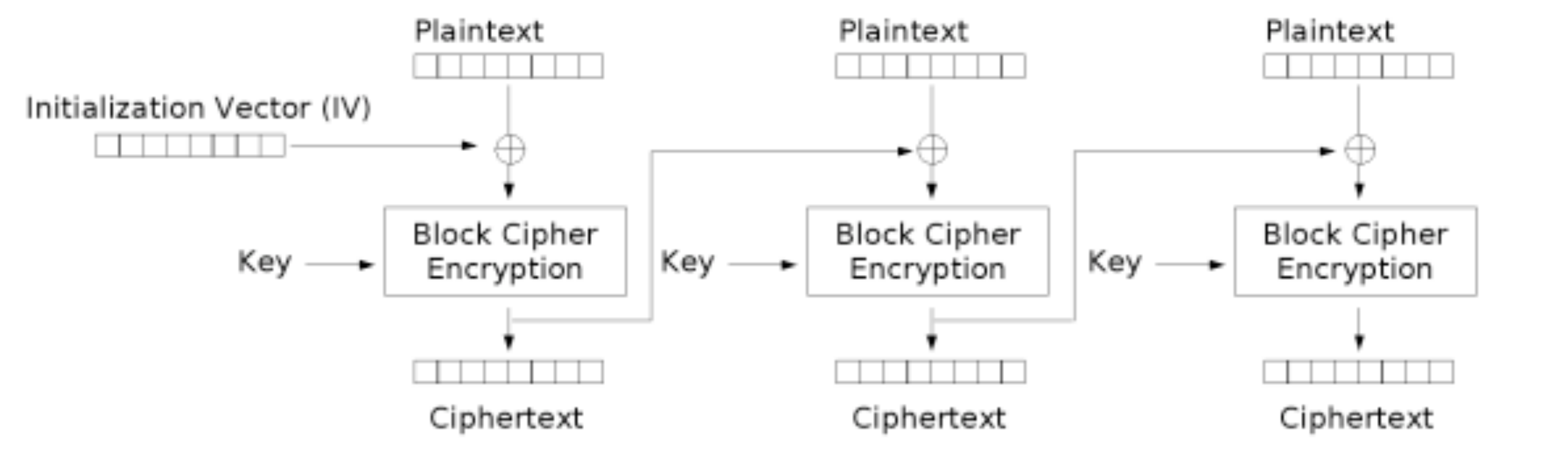
- Assignment 1 is out
- Course Project:
    - Type of Projects
        - Survey
        - Exploration
    - Proposals will be due in a few weeks
    - Teams of 2-4
    - In-Class Presentation + Writeup
    - We will share some topic ideas
    - Will summarize all of this in a document

News?

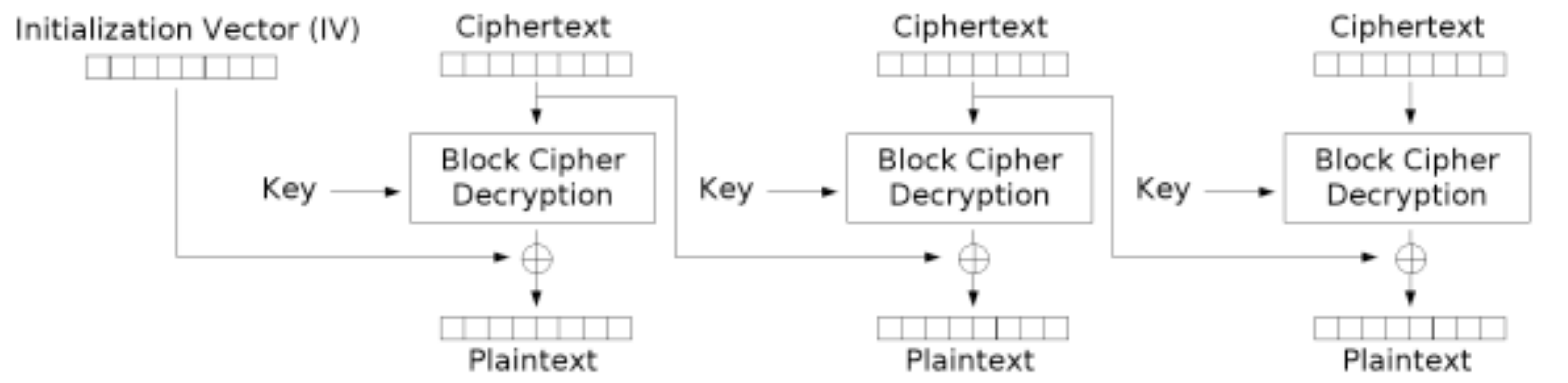
Using Block Ciphers

- ECB is not semantically secure, hence we use a "mode of operation"
  - e.g., CBC, CTR, CFB, OFB (and others)
- These provide:
  - Security for multi-block messages
  - Randomization (through an Initialization Vector)

CBC Mode



Cipher Block Chaining (CBC) mode encryption



Cipher Block Chaining (CBC) mode decryption

Public domain image courtesy Wikipedia.

Security of CBC

- Is CBC a secure encryption scheme?
  - Yes, assuming a secure block cipher
  - Correct (random) IV generation
  - Can <u>prove</u> this under assumption that
    block cipher = Pseudo-Random Permutation (PRP)
- Bellare, Desai, Jokipii & Rogaway (2000)
- Easy to use wrong...
  - Most important: use a unique & random IV!

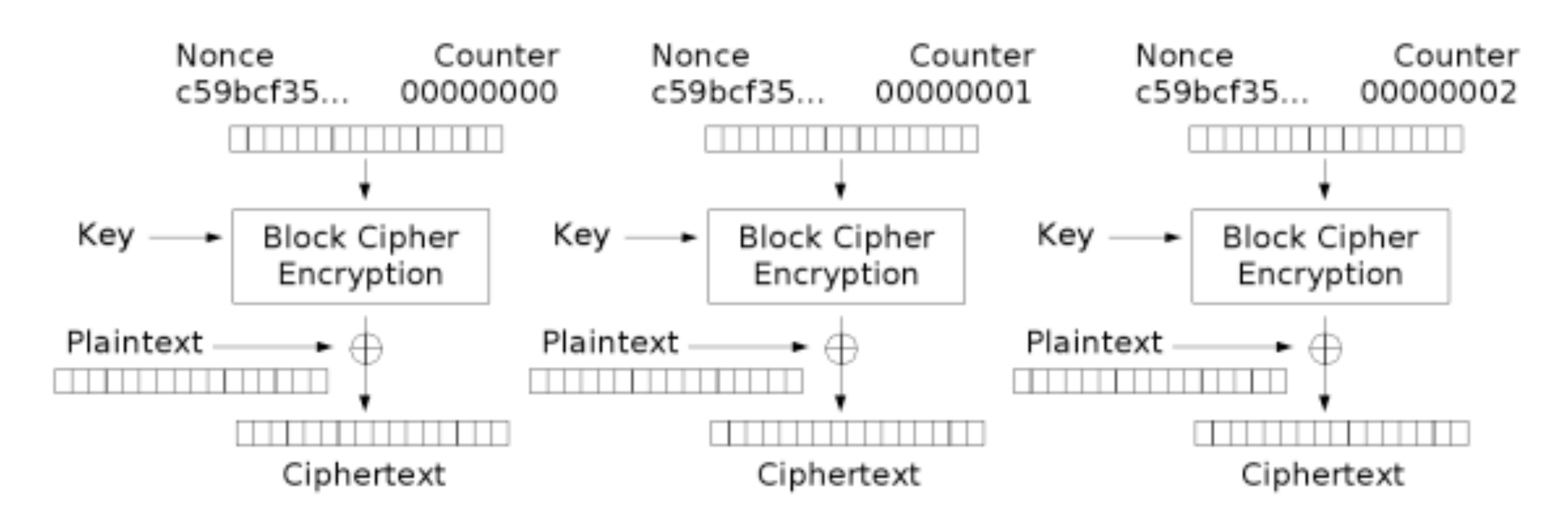
The size of the frame of data to be encrypted or decrypted (i.e. how often a new CBC chain is started) depends on the particular application, and is defined for each in the corresponding format specific books of this specification. Unless otherwise specified, the Initialization Vector used at the beginning of a CBC encryption or decryption chain is a constant, iv_0, which is:

$$0BA0F8DDFEA61FB3D8DF9F566A050F78_{16}$$

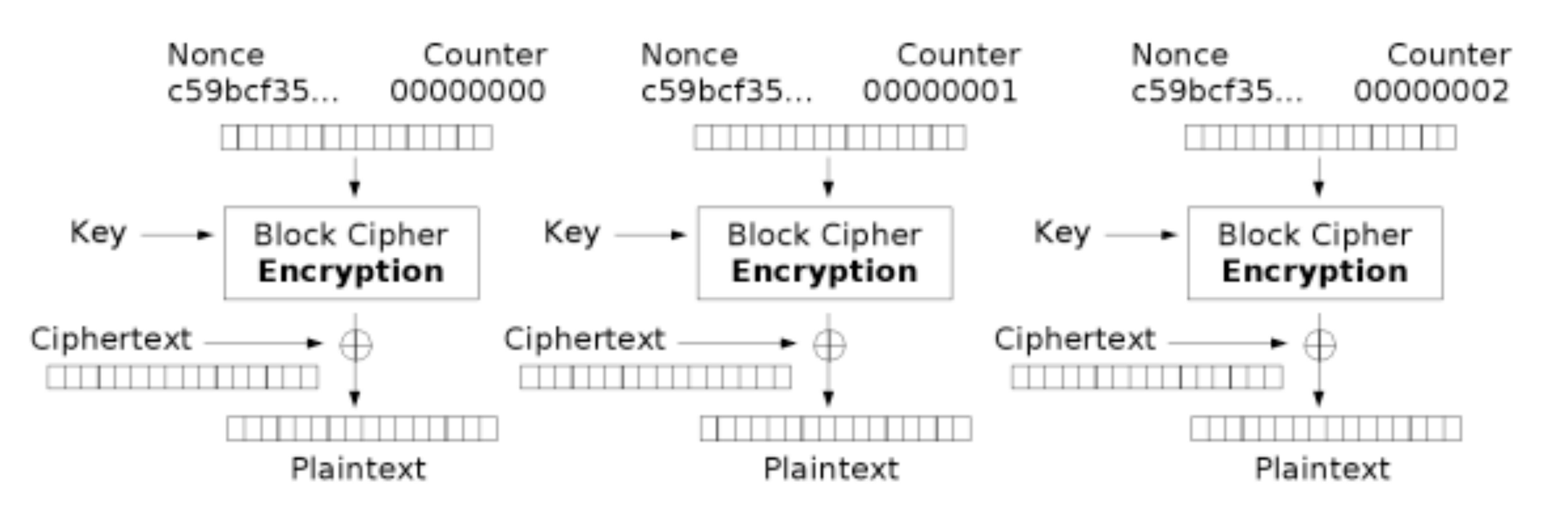
Advanced Access Content System (AACS)

Introduction and
Common Cryptographic Elements

CTR Mode



Counter (CTR) mode encryption



Counter (CTR) mode decryption

Public domain image courtesy Wikipedia.

Security of CTR

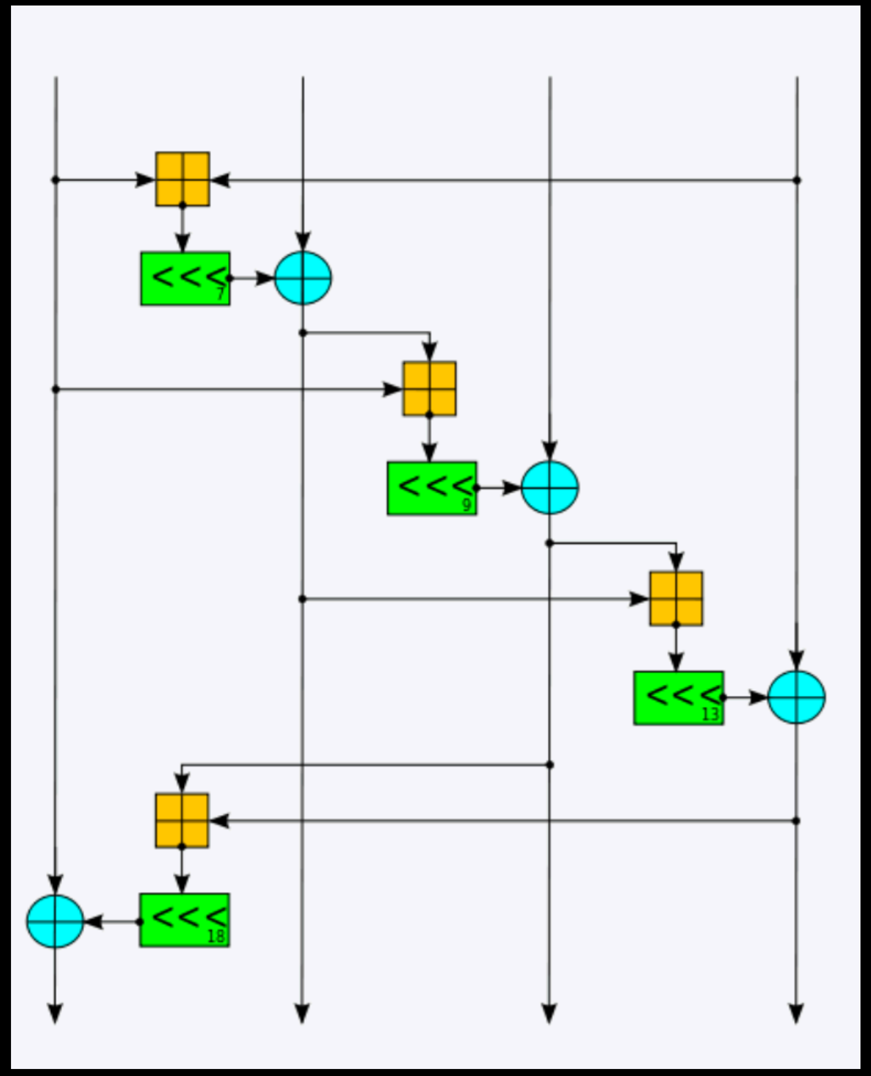
- Yes, assuming secure block cipher (PRP)
- However, counter range must <u>never</u> be re-used

$$\begin{matrix}\text{Plaintext 1}\\ \oplus \\ \cancel{\text{Keystream}}\end{matrix} \oplus \begin{matrix}\text{Plaintext 2}\\ \oplus \\ \cancel{\text{Keystream}}\end{matrix} = \begin{matrix}\text{Plaintext 1}\\ \oplus \\ \text{Plaintext 2}\end{matrix}$$

- Similar example: MS Word 2003
  - (they used RC4, but same problem)

Alternative ciphers

- Salsa20, ChaCha (Bernstein)
  - These are <u>not</u> block ciphers
  - Designed as *non-invertible pseudorandom* functions
  - Salsa20(k, n) -> {output}
  - Can use these to implement a stream cipher (i.e. CTR mode)



Point of order

- Proofs of security:
  - We don't know how to prove that DES or AES or Salsa20 are secure block ciphers
  - But if we assume that the block ciphers are secure PRPs (resp PRFs) then:

- We can prove that CBC & CTR & OFB & CFB etc. are secure encryption modes.

http://www.cs.ucdavis.edu/~rogaway/papers/sym-enc-abstract.html

Point of order

In 2008, Bernstein published the closely related "**ChaCha**" family of ciphers, which aim to increase the diffusion per round while achieving the same or slightly better performance.[17] The Aumasson et al. paper also attacks ChaCha, achieving one round fewer: for 256 bits ChaCha6 with complexity 2^{139} and ChaCha7 with complexity 2^{248}. 128 bits ChaCha6 within 2^{107}, but claims that the attack fails to break 128 bits ChaCha7.[3]

- Proofs
  - We … block ciph…
  - But if we assume that the block ciphers are secure PRPs (resp PRFs) then:

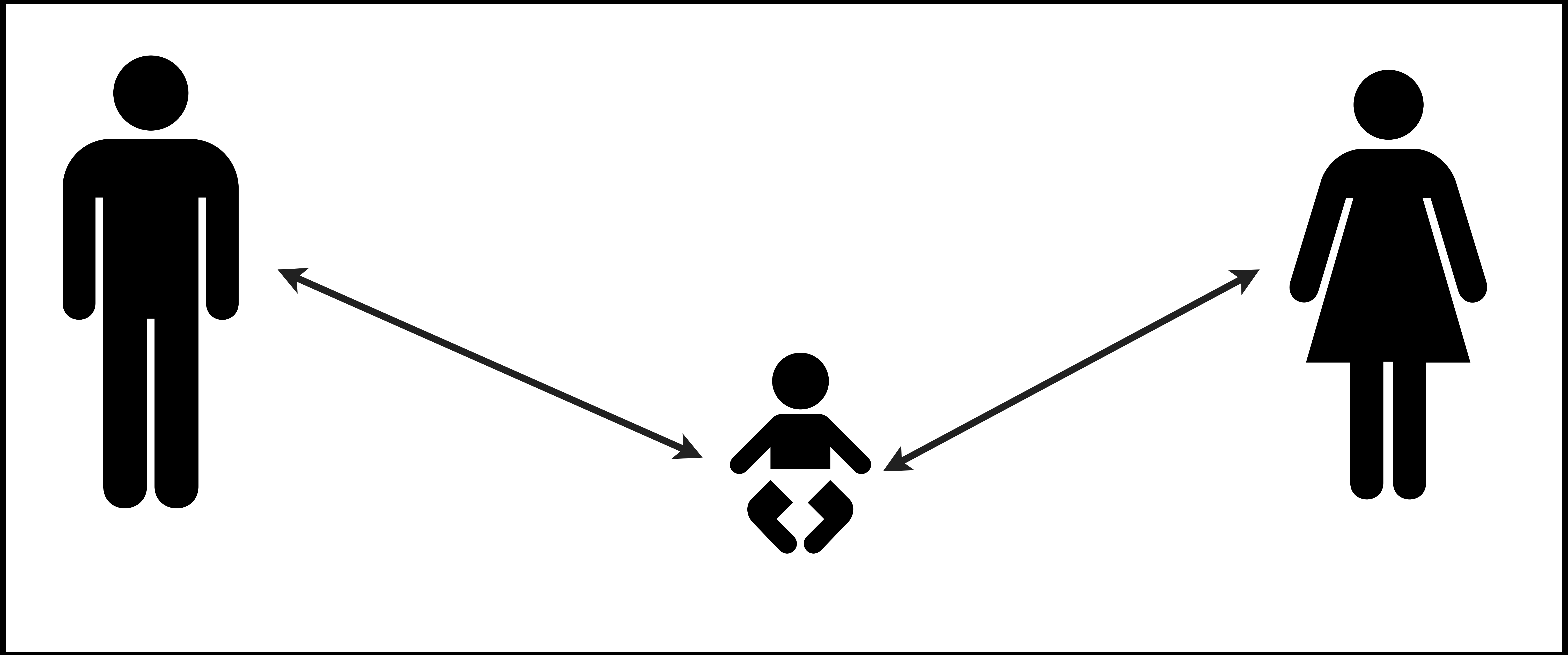
- We can prove that CBC & CTR & OFB & CFB etc. are secure encryption modes.

http://www.cs.ucdavis.edu/~rogaway/papers/sym-enc-abstract.html

Malleability

- The ability to modify a ciphertext
  - Such that the plaintext is meaningfully altered
  - CTR Mode (bad)
  - CBC Mode (somewhat bad)

Authenticated Encryption



MACs

- Symmetric-key primitive
  - Given a key and a message, compute a “tag”
  - Tag can be verified using the same key
  - Any changes to the message detectable
- To prevent malleability:
  - Encrypt <u>then</u> MAC
  - Under separate keys

MACs

- Definitions of Security
  - Existential Unforgeability under Chosen Message Attack (EU-CMA)
- Examples:
  - HMAC (based on hash functions)
  - CMAC/CBC-MAC (block ciphers)

Authenticated Encryption

- Two ways to get there:
  - <u>Generic composition</u>
    Encrypt (e.g., CBC mode) then MAC

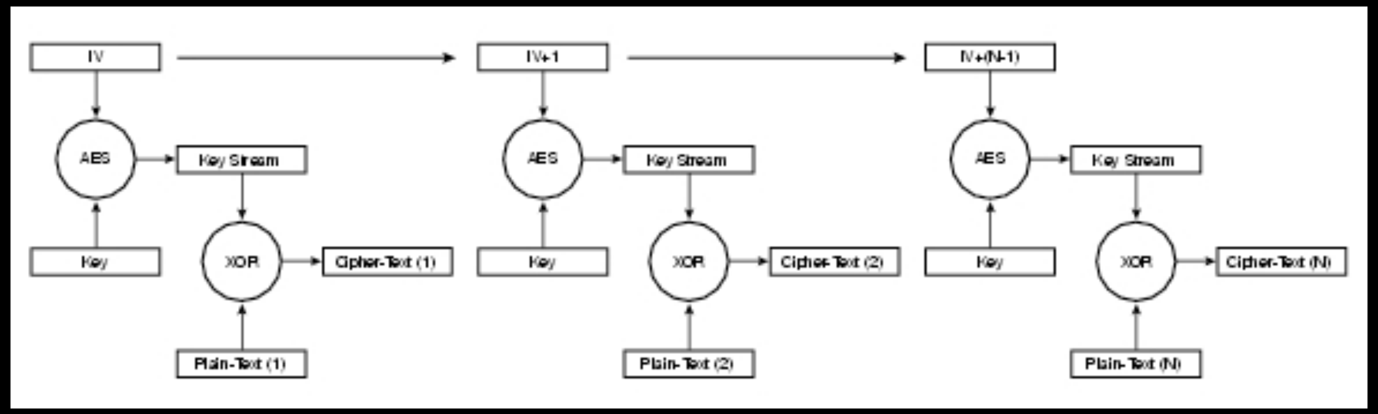
•two different keys, multiple primitives

  - <u>Authenticated</u> mode of operation

•Integrates both encryption & authentication

•Single key, typically uses only one primitive (e.g., block cipher)

•Ex: CCM, OCB, GCM modes



ZDNet

December 30th, 2008

SSL broken! Hackers create rogue CA certificate using MD5 collisions

Posted by Ryan Naraine @ 6:00 am

Categories: Zero-day attacks, Microsoft, Browsers, Punditocracy, Responsible disclosure...
Tags: Certification Authority, SSL, Web Browser, Computer Associates International Inc., Certificate...

71 TalkBacks ADD YOUR OPINION | SHARE | PRINT | E-MAIL | WORTHWHILE? +129 141 VOTES

Using computing power from a cluster of 200 PS3 game consoles and about $700 in test digital certificates, a group of hackers in the U.S. and Europe have found a way to target a known weakness in the MD5 algorithm to create a rogue Certification Authority (CA), a breakthrough that allows the forging of certificates that are fully trusted by all modern Web browsers.

SSL is not broken: The facts surrounding the CCC disclosure

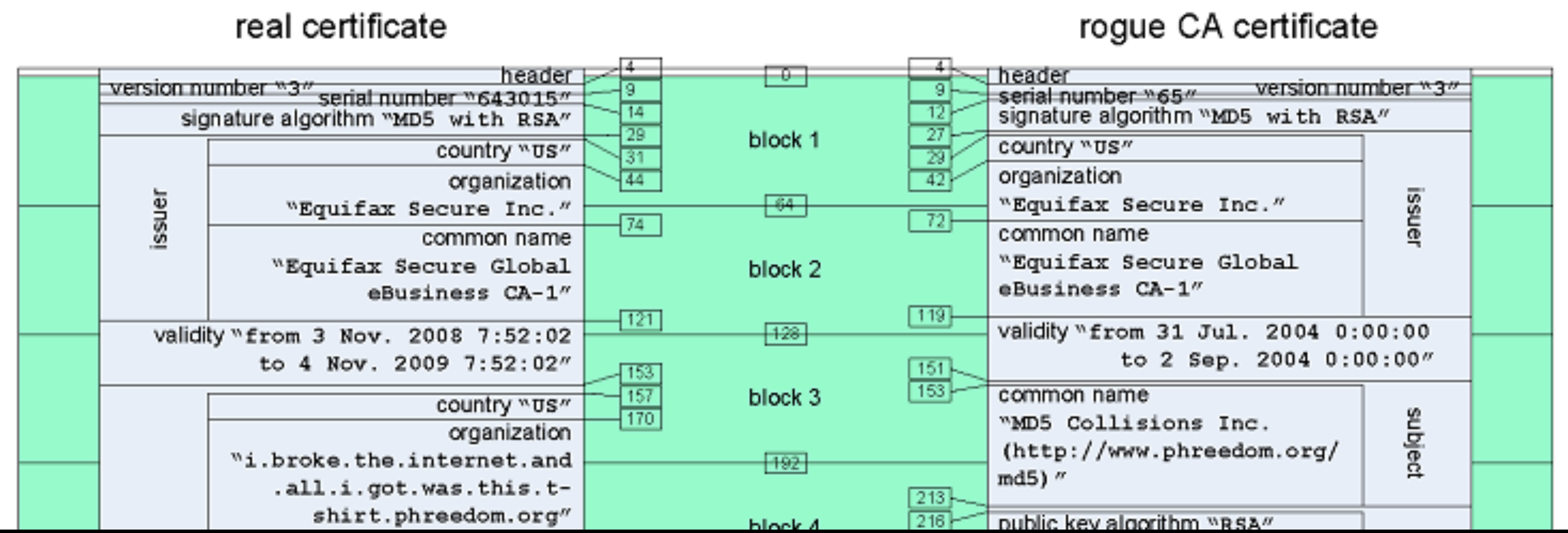
by Steve Ragan – Jan 6 2009, 11:00

MD5 considered harmful today

Creating a rogue CA certificate

December 30, 2008

Alexander Sotirov, Marc Stevens,
Jacob Appelbaum, Arjen Lenstra, David Molnar, Dag Arne Osvik, Benne de Weger



Hash Functions

- Convert variable-length string to small “tag”
  - Hash tables
  - Signatures
  - Software checksums
  - MAC functions (HMAC)
  - Encryption (OAEP)

Cryptographic

Hash Checksums

```
Debian GNU/Linux 4.0 alias etch
- -------------------------------

Source archives:

  http://security.debian.org/pool/updates/main/l/lighttpd/l...
    Size/MD5 checksum:     1108 d747ed7b2063ad6696064bf821c50a00
  http://security.debian.org/pool/updates/main/l/lighttpd/l...
    Size/MD5 checksum:    38244 c6de19903fcf9972a3db86af50c3dfb6
```

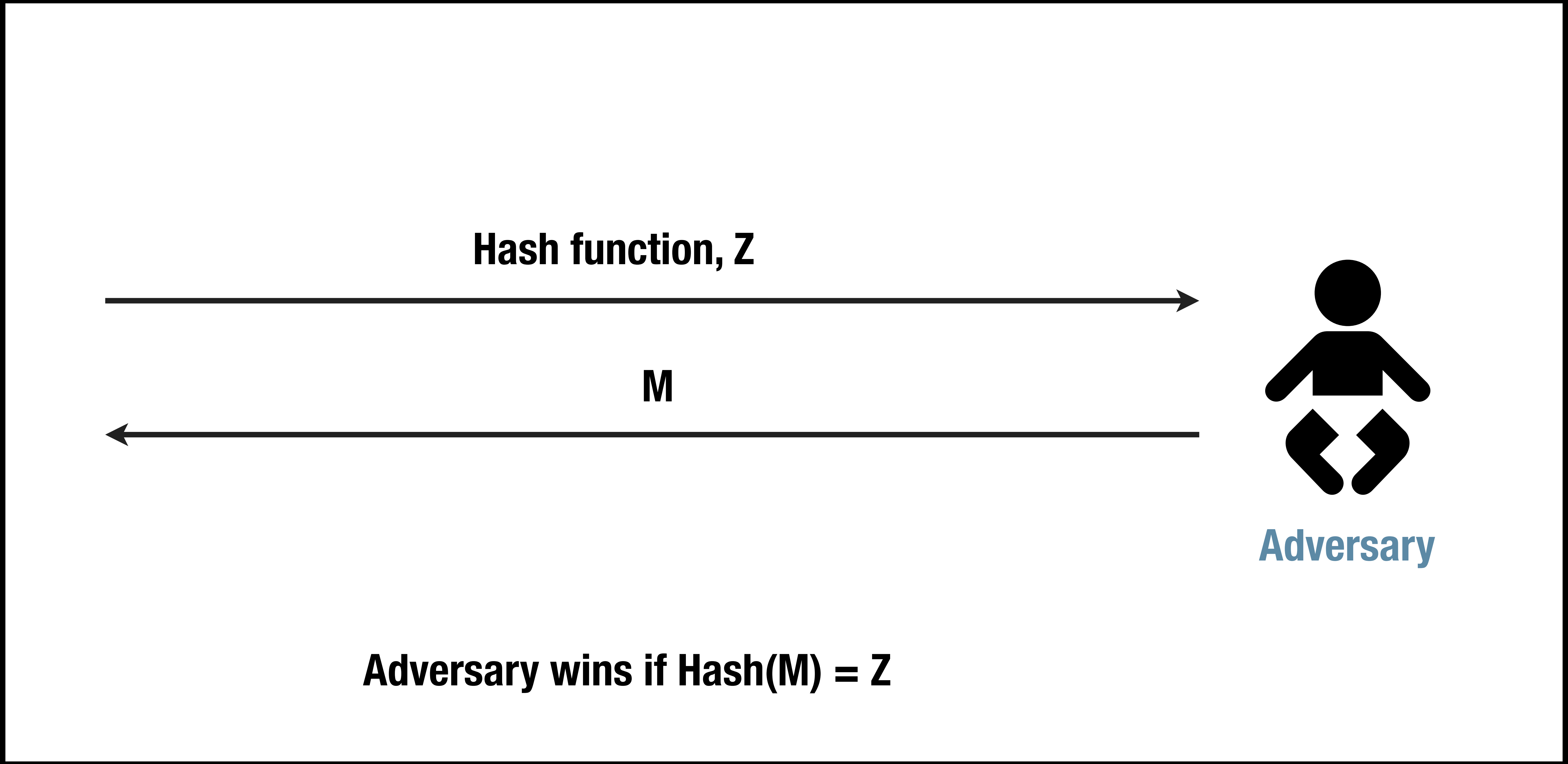
Cryptographic Hashing

- What, exactly, do we require?
- We have some guidelines:
  - Efficiency
  - Pre-image resistance
  - Collision Resistance
  - Second Pre-Image Resistance

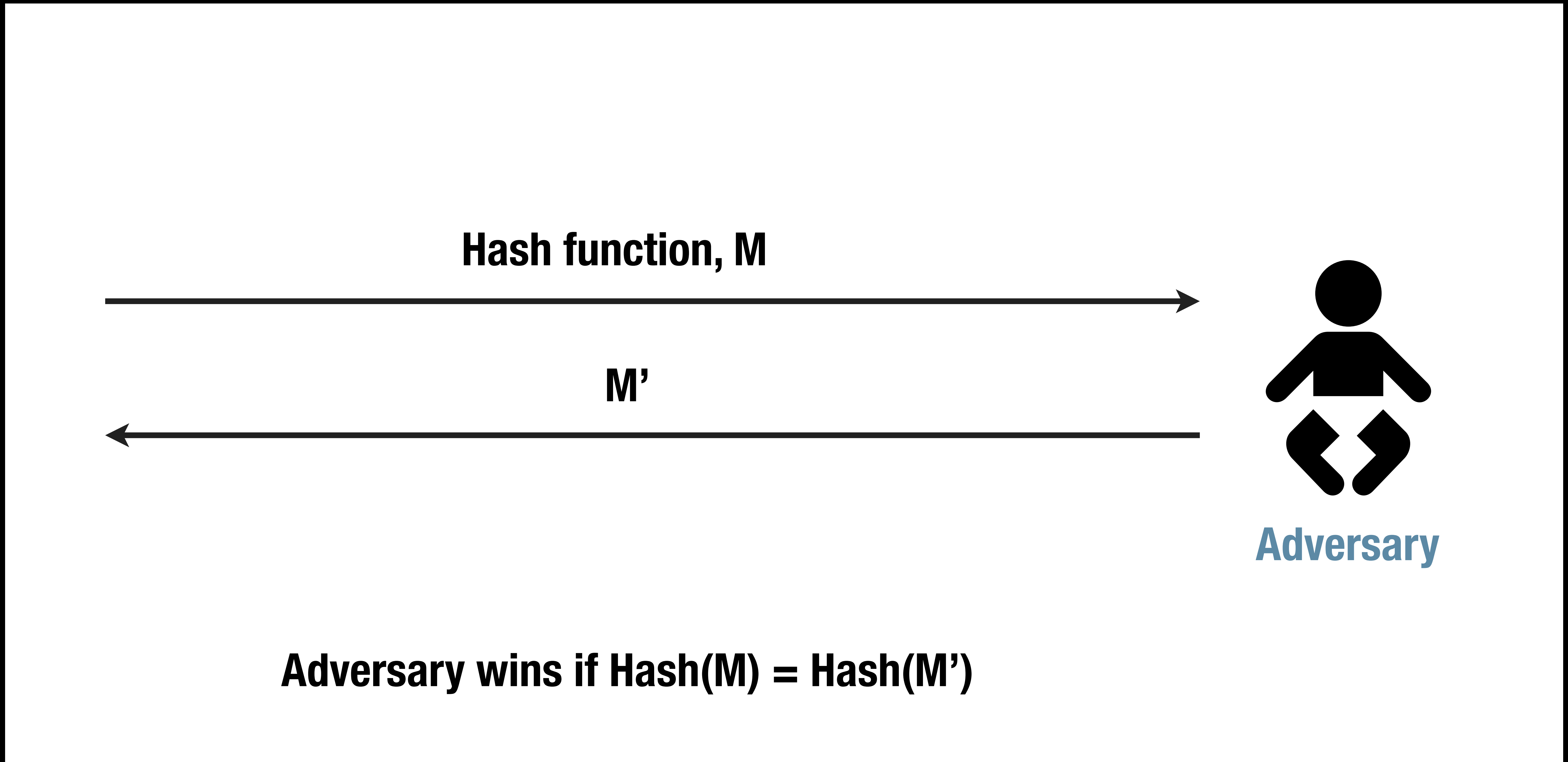
Efficiency

- Efficient to compute
- Algorithm is compact
- Theoretical definition:
  - computable in polynomial time (of input size)
  - this implies polynomial-size description

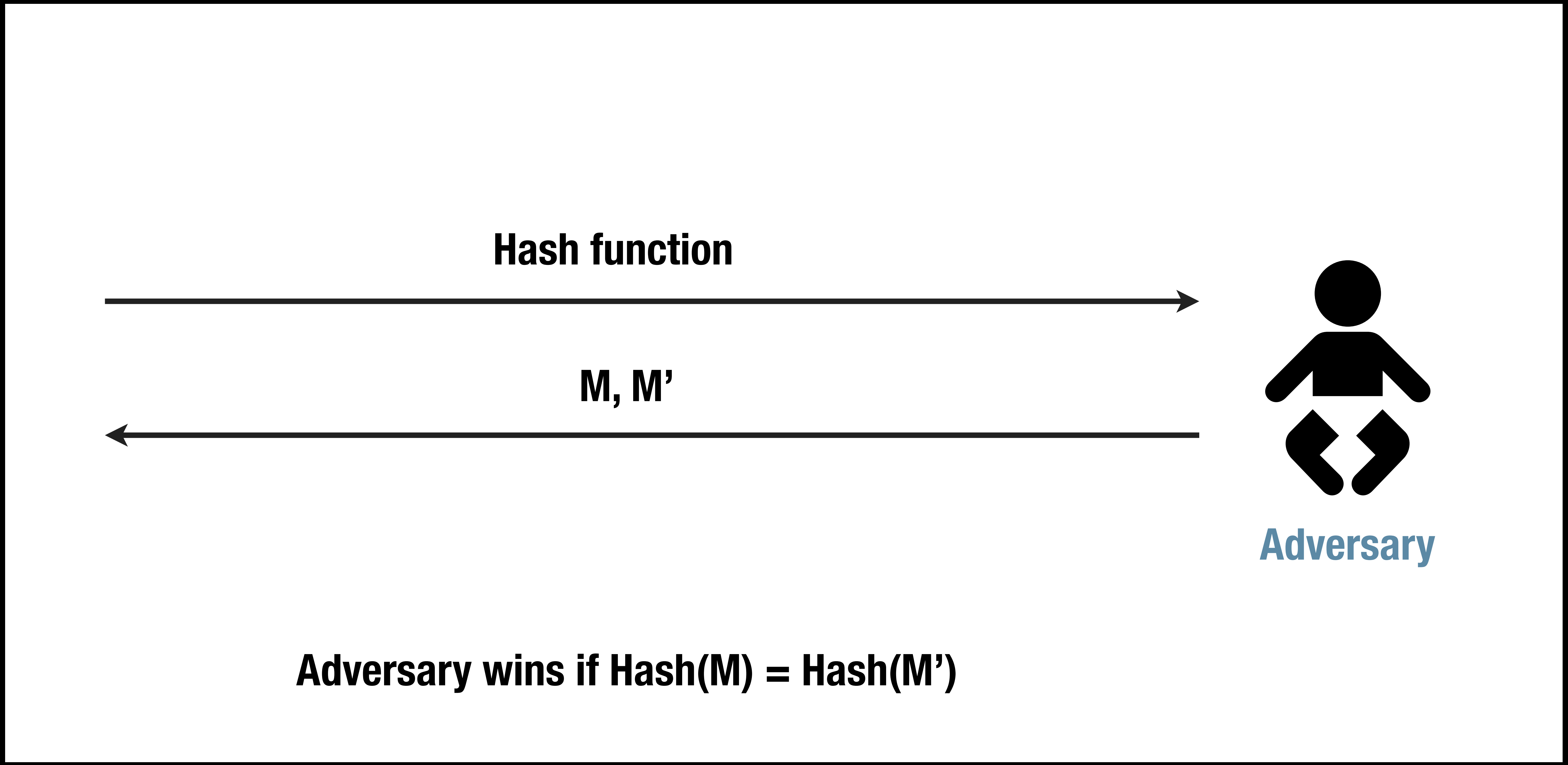
Pre-image Resistance



2nd Pre-image Resistance



Collision Resistance

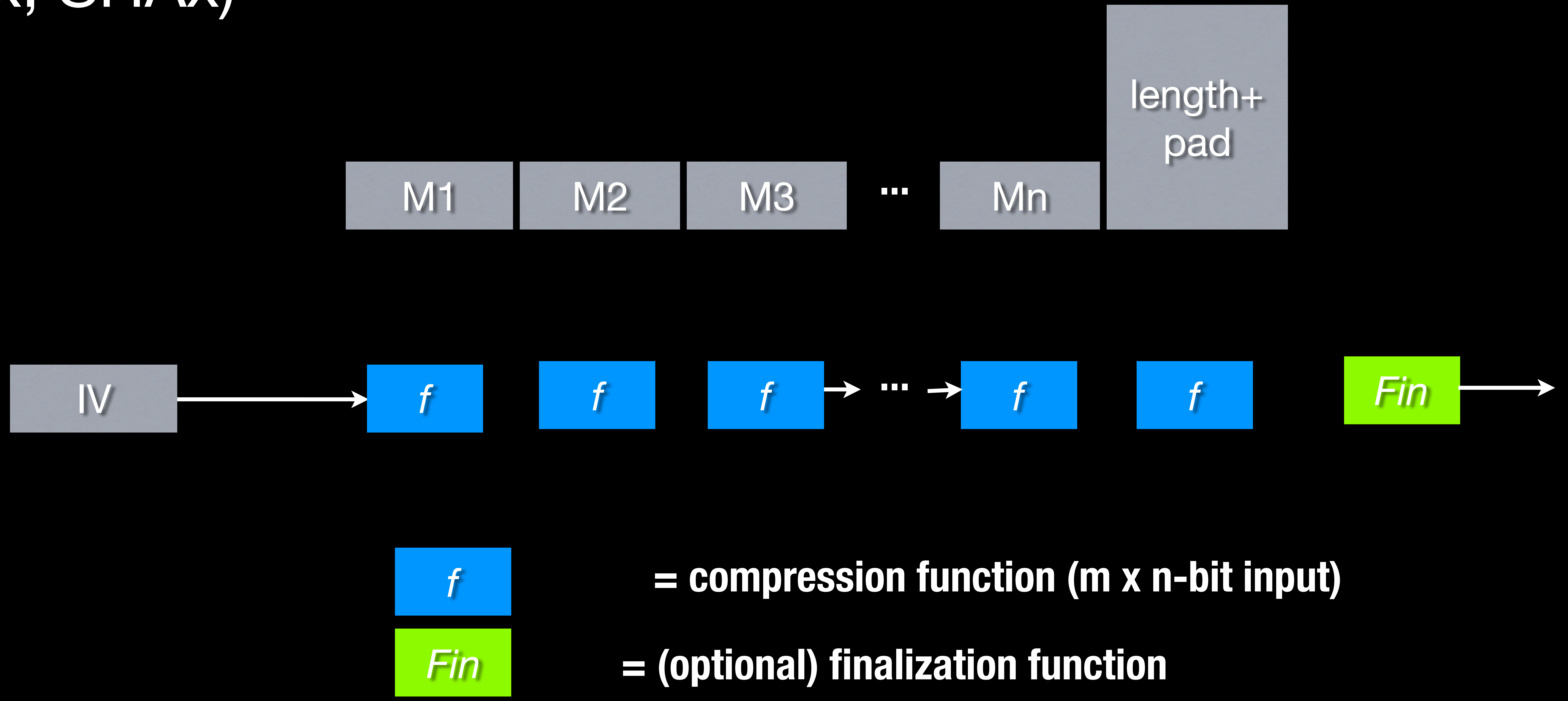


Ideal Hash Function

- What would a perfect hash function look like?
  - Outputs completely unrelated to inputs
  - E.g., a random function
- So...
  - H(M) leaks no special information about M
  - Collisions & 2nd preimages hard to find

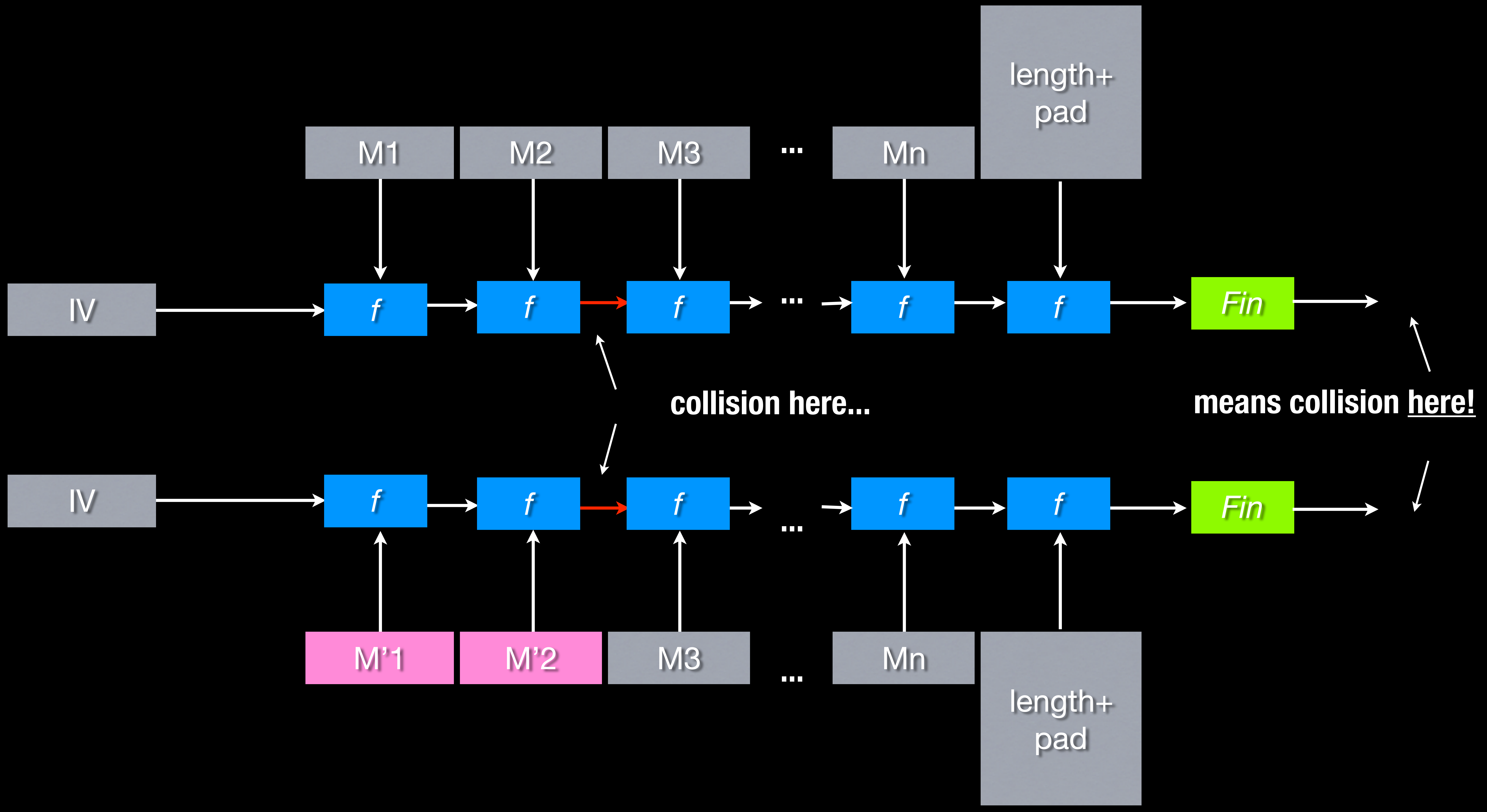
Merkle-Dåmgard

- Used in most standard hash functions
  - (MDx, SHAx)

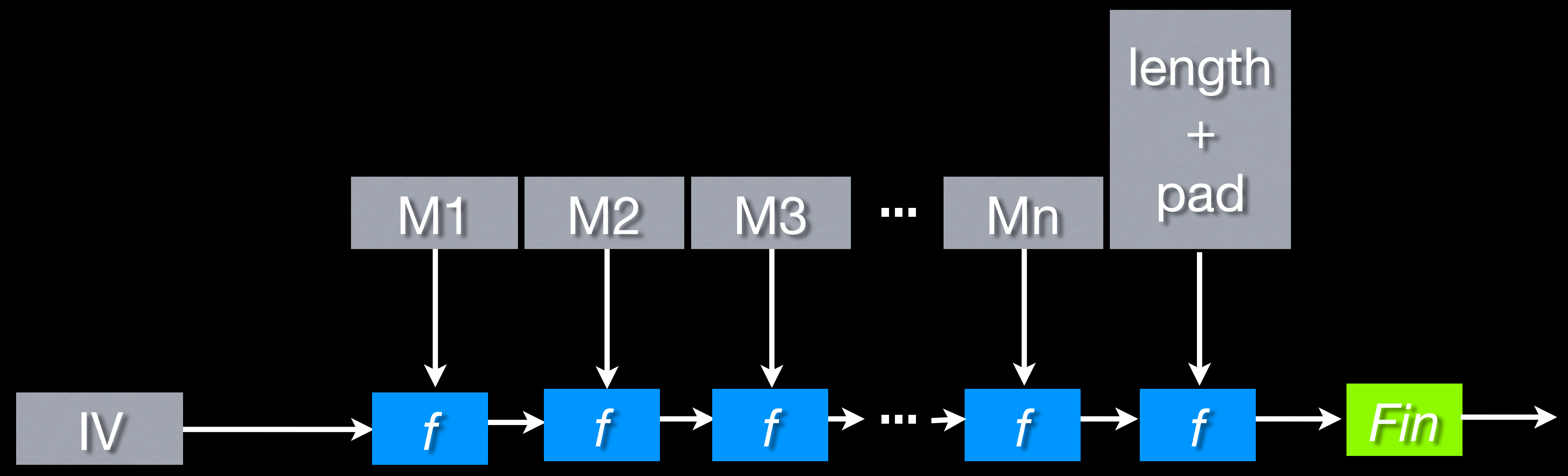


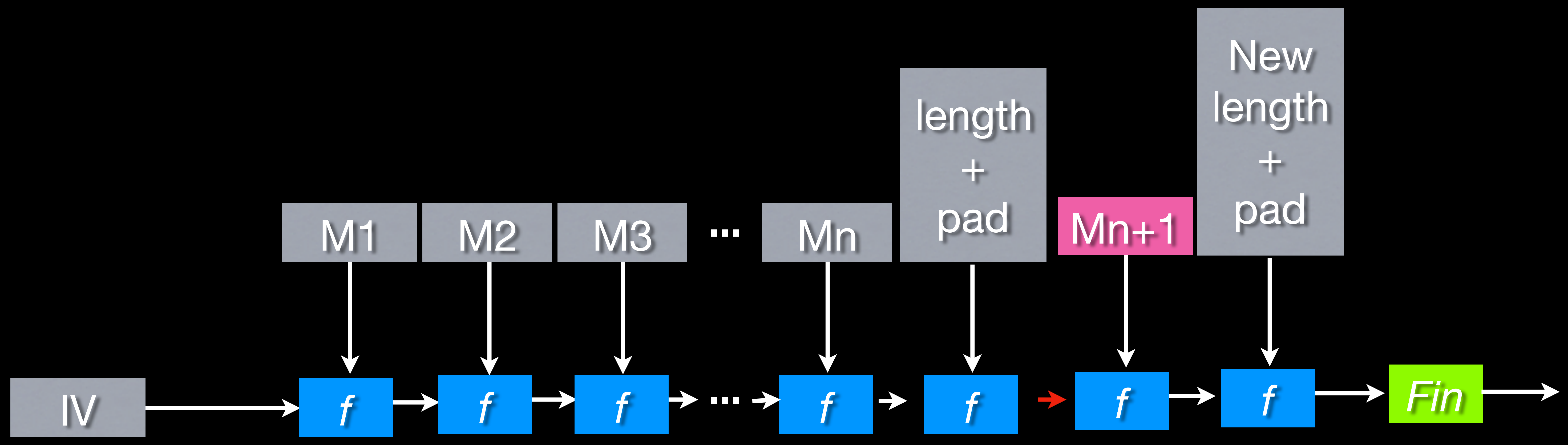
- Why Merkle-Damgard?
  - If f is collision-resistant, then H() is too (Crypto '89)
  - If f is an ideal cipher (random function), then H() is an ideal hash function

Collision Attacks



Length Extension Attacks





Recent Collision Attacks

Image of Xiaoyun Wang from Shandong University website: http://www.infosec.sdu.edu.cn/

Google Security Blog

The latest news and insights from Google on security and safety on the Internet

Announcing the first SHA1 collision

February 23, 2017

Posted by Marc Stevens (CWI Amsterdam), Elie Bursztein (Google), Pierre Karpman (CWI Amsterdam), Ange Albertini (Google), Yarik Markov (Google), Alex Petit Bianco (Google), Clement Baisse (Google)