Security Basics

Ruby B. Lee
Princeton University
HotChips Security Tutorial
August 10 2014

Outline

- What is Security
- Threat Model
- Security Design Methodology
- Security Policies
- Access Control
  - Authentication and Authorization
- Cryptography
- Security Protocols

What is Security?

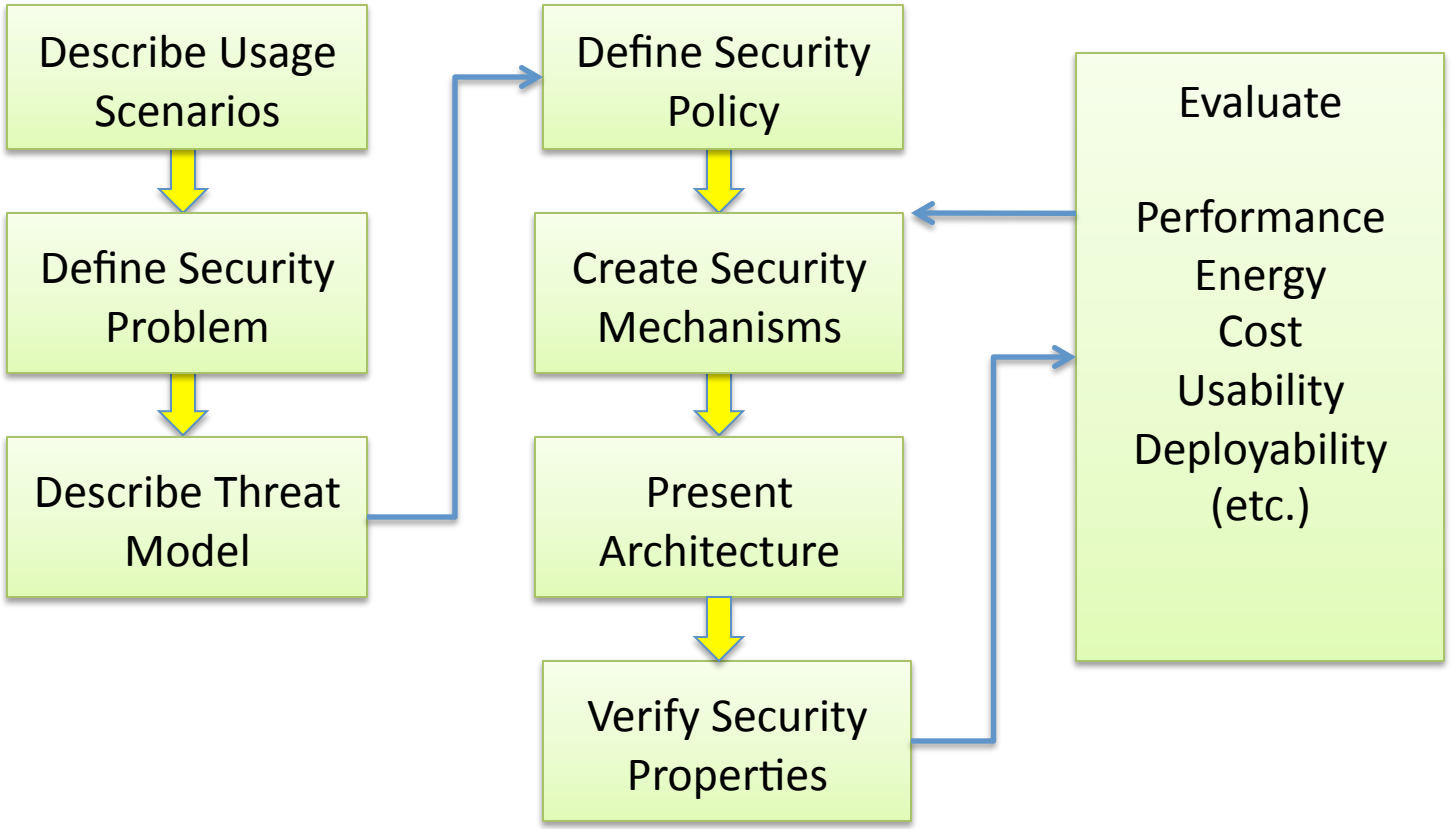
Cornerstone security properties (CIA):

- Confidentiality
  - Prevent disclosure of information to an unauthorized entity
    - Example Attack: Eavesdropping
- Integrity
  - Prevent unauthorized modification without detection
    - Example Attack: Corruption attacks (code injection)
- Availability
  - System and services are available when requested by legitimate users
    - Example Attack: Denial of Service

More Security Aspects

- Access control policy
  - Specifies which principals can access which objects/resources
  - Authentication and Authorization are essential aspects
- Accountability and Attribution
  - Specifies if/how actions can be tied to attacks
- Non-repudiation
  - Ability to hold one responsible to messages/events
- Anonymity
  - Ability to carry out actions without identification
- Privacy
  - Right to determine how one's personal information is distributed
- Distinction between Security (confidentiality) & Privacy
  - Confidentiality is the obligation to protect secret information
  - Privacy is the right to protect distribution of personal information

Lee's Security Architecture Design Methodology



Threat Model

- A threat model defines
  - the threats that are being considered
  - the threats that are not being considered
    - e.g., Consider threats/attacks on the confidentiality and integrity of sensitive data, but not Denial of Service threats/attacks.
  - the basic assumptions of the computing model
- An attack is an instantiation of a threat
- Attacks violate security properties of a system

Threat-based design

- Threat-based design is key difference for hardware designers
- Each speaker will talk about different threats & attacks
- System vulnerabilities exploited in attacks
  - Several vulnerabilities databases exist, e.g., http://cwe.mitre.org
- Arm's race: no such thing as absolute security
- Difference between Security and Reliability
  - Smart malicious attackers versus failures with characteristic/statistical behavior

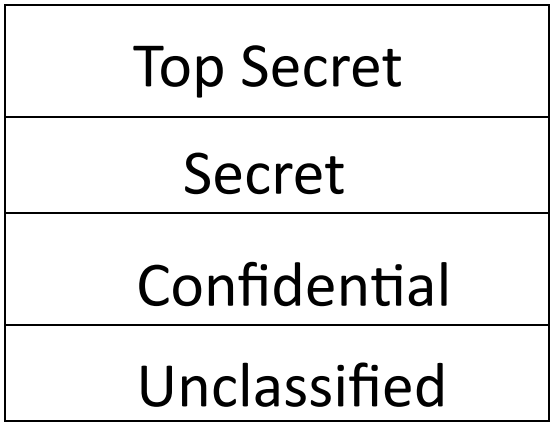
Security Policies vs. Security Mechanisms

- Security Policy
  - specifies what and who is allowed access to what resources or information, when.
- Security mechanisms
  - implements the security policy

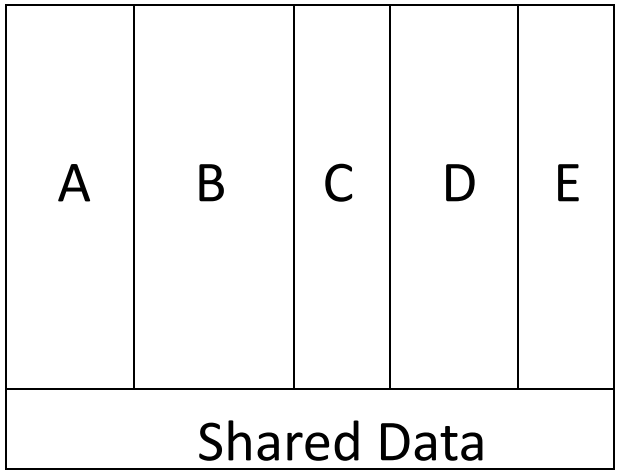
- Trusted vs. Trustworthy
  - Trusted – depended on to maintain the security policy
  - Trustworthy – designed to be secure, dependable
  - a trusted component may not be trustworthy, and vice versa

- Trusted Computing Base (TCB)
  - the hardware, software or networking components that must be correct and un-corruptible, otherwise the security policy may not hold

Multilevel vs. Multilateral Security Policies



Multi Level Security (MLS)



Multi-lateral Security, or
Compartmented Security

MLS: Hierarchical Security Levels

- Military (DoD)
  - Subjects have clearances
  - Objects have classifications
- **TS** > **S** > **C** > **U partial ordering**

Commercial

Top Secret, TS
Secret, S
Confidential, C
Unclassified, U

CEO and VPs
Managers
Employees
Non-employees

- MLS is a form of Mandatory Access Control (MAC)
  - Needed when subjects/objects at different security levels use same system

Bell-LaPadula (BLP)

- BLP protects confidentiality
  - To prevent unauthorized reading
- BLP rules:
  - No subject may read data classified at a higher security level
    - Simple Security property:
    - A subject can read an object only if his clearance level is equal to or greater than the object's classification level.
  - No subject may write data to a lower security level
    - *-Property:
    - A subject can write an object only if his clearance level is less than or equal to the object's classification level.
- No read up, no write down

Why *-Property?

- Consider the "corrupted general"
- Prevents information leakage by a higher security subject, S1, to a lower security subject, S2.
  - Let clearance of General Smith = TS and clearance of Vladimir = U
- If no *-property, General Smith can read a TS file X and copy it to another file Y at U level that Vladimir can read.

Other Security Policy Models

Multi-Level Security Policy	Multi-Lateral Security Policy
BLP (Confidentiality)	Chinese Wall (Confidentiality)
Biba (Integrity)	Clark Wilson (Integrity)
MLS with Codewords	British Medical Assoc (BMA)
…	…

- Security policy models help us understand basic requirements for confidentiality and integrity, etc.
- Real-world security policies can be very complicated, e.g., HIPAA, etc.
  - translated into specific policy languages or XML, if policy is evaluated by computer for Authorization or Compliance purposes.

Access Control (AAA)

- **Authentication: Who are you?**
  - Authenticate human/machine to machine
  - What you know, What you have, What you are.
- **Authorization: What are you allowed to do?**
  - Restrict actions of authenticated users
  - Who can do what to which object?

    (Subject, object, rights)

	Final Grades	Homework Grades	Exam Prep	Lecture Slides
Professor	R, W	R	R, W	R,W
TA	R	R, W	R, W	R
Student	-	-	-	R

Access Control Matrix

Types of Access Control

Type of Access Control	Access determined by:	Requires:
Mandatory Access Control (MAC)	System	Trusted system; with *Rule-based access control*
Discretionary Access Control (DAC)	User	Authentication of user; with *Identity-based access control*
Role-based access control (RBAC)	Current role of user	Authenticating user and verifying his current role
Originator-based access control (ORCON)	Originator (or creator)	"Policy specified by originator must be enforced on associated item forever."

Outline

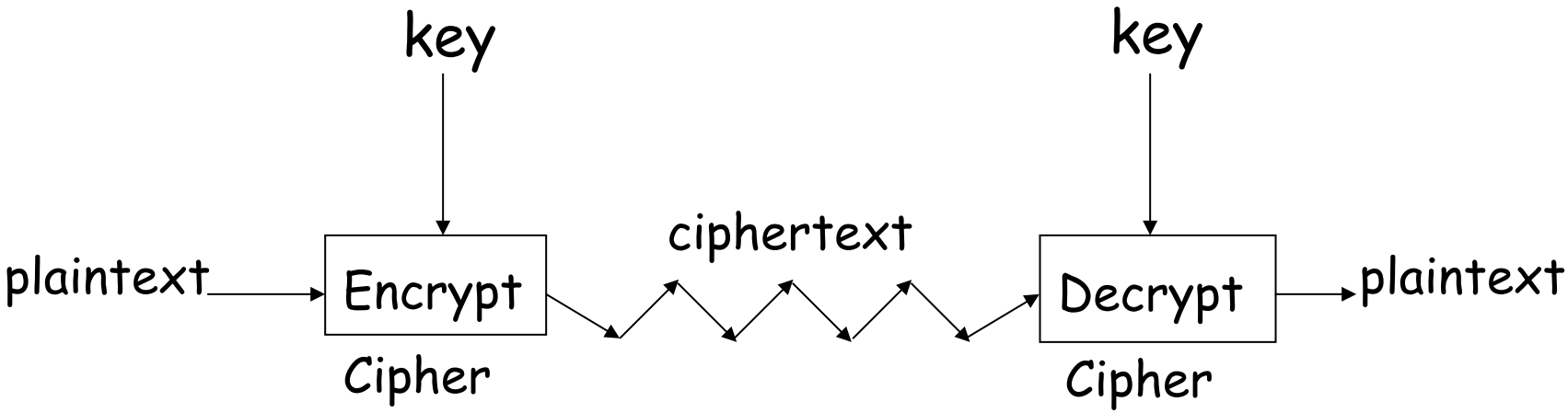
- What is Security
- Threat Model
- Security Design Methodology
- Security Policies
- Access Control
  - Authentication and Authorization
- **Cryptography**
  - **Symmetric-key Crypto, Cryptographic Hash, Public-Key Crypto and PKI**
- Security Protocols

Classes of Cryptography

- Symmetric Key Ciphers
  - Useful for protecting Confidentiality
- Cryptographic Hash functions
  - Useful for protecting Integrity
- Public Key Ciphers
  - Useful for longer-term identity: authentication, digital signatures and non-repudiation

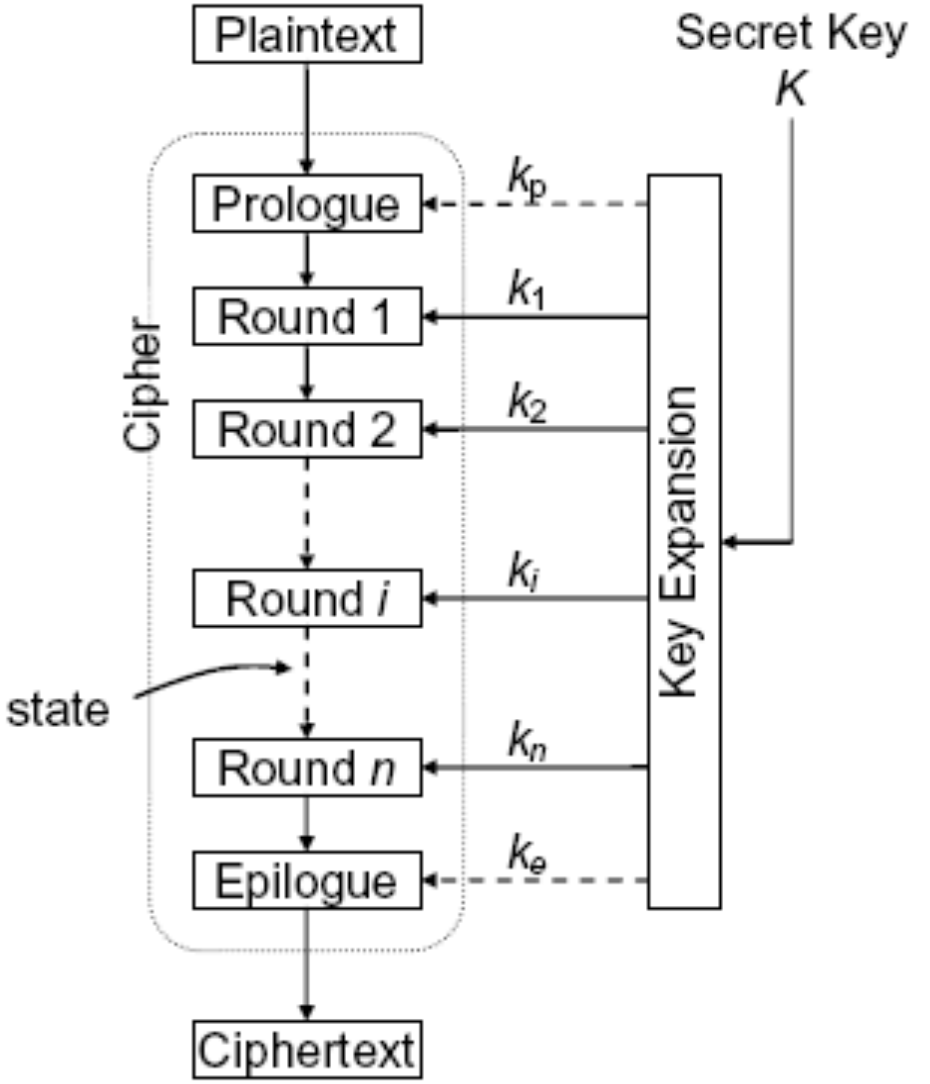
Cryptographic access control: put encrypted data in public, but control access to keys

Cryptography



- Symmetric-key crypto
  - Shared secret key used for encryption and decryption

Block Ciphers



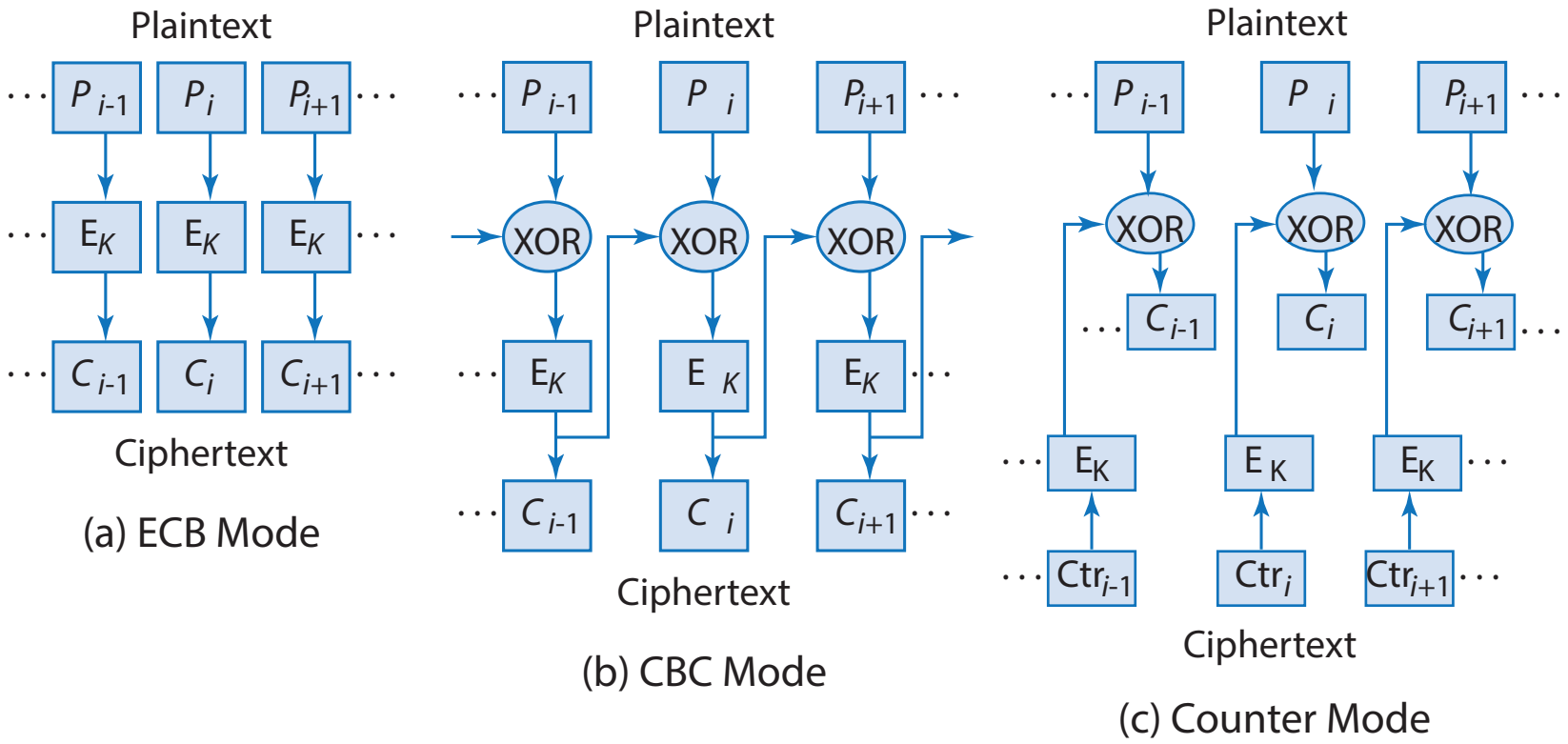
Examples:

AES = Advanced Encryption Std.

DES = Data Encryption Std.
3DES = Triple DES

Many others

Modes of Operation



Electronic Code Book Mode

Cipher Block Chaining Mode

Counter Mode

Cryptographic Hash

- Acts like a fingerprint of a message
- h = hash(M)
  - "Bob Smith got an A+ in ELE386 in Spring 2005" → 01eace851b72386c462de6ba6ec76f68
  - "Bob Smith got an B+ in ELE386 in Spring 2005" → 936f8991c111f2cef61e90db0650ef4d

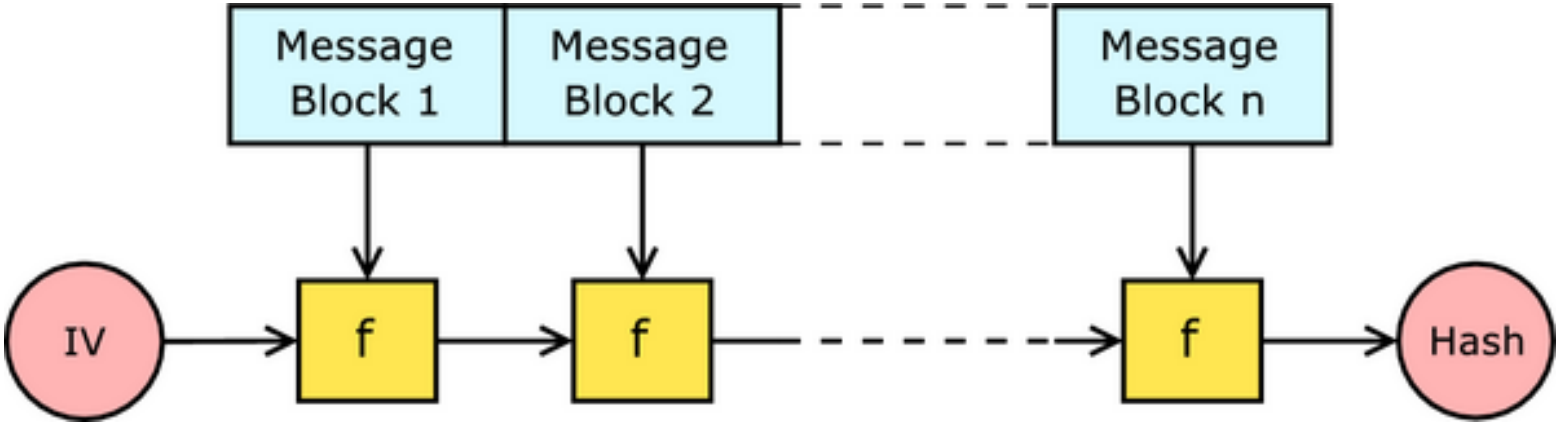
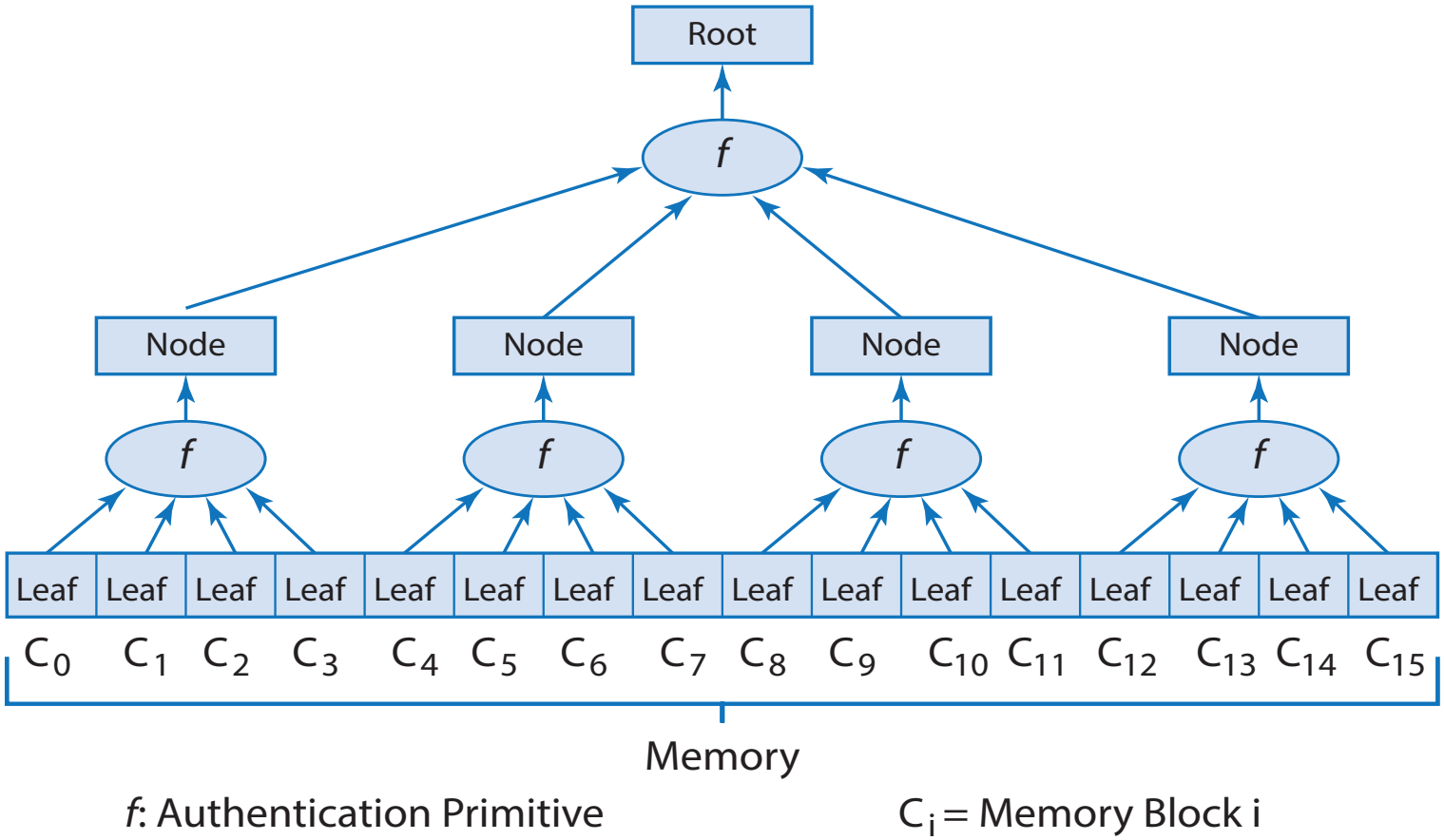


Diagram: http://en.wikipedia.org/wiki/Cryptographic_hash_function

Cryptographic Hash Uses

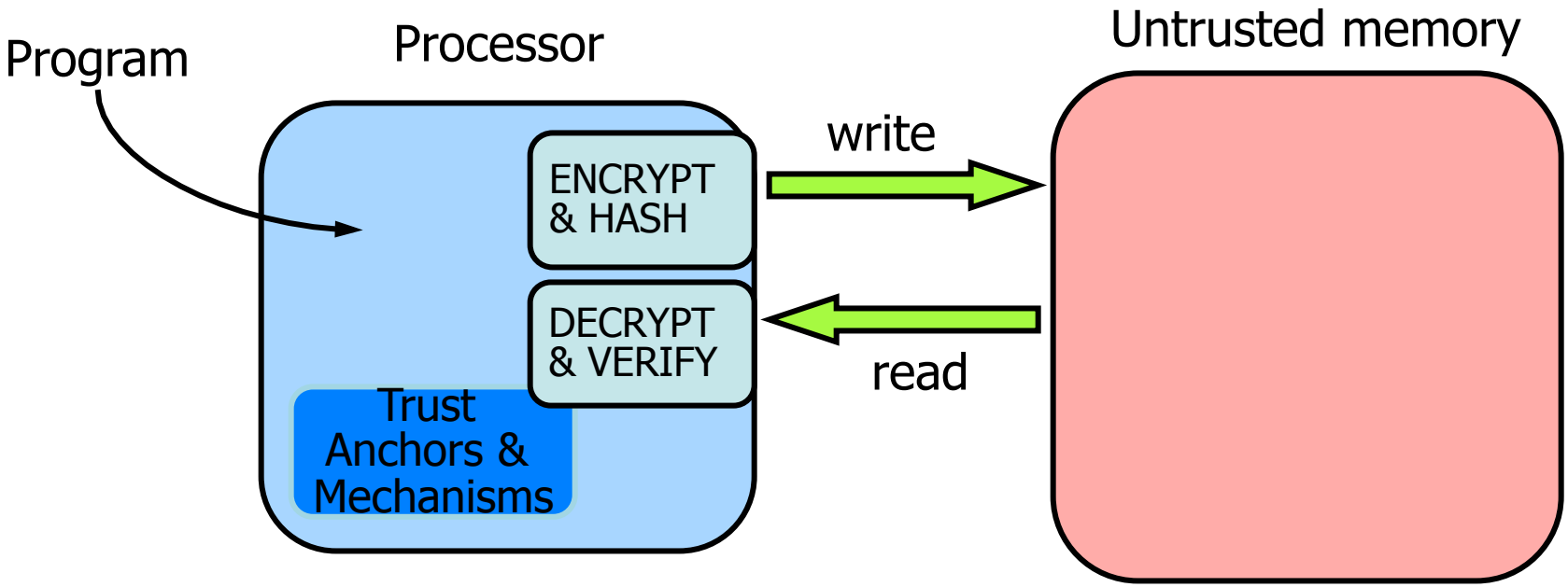
- Useful for Integrity
- Data/Code corruption detection
- Message fingerprint
- Identity of a software module
- Keyed hash (with secret key)
  - Message Authentication Code (MAC, e.g., HMAC)
- Digital signature efficiency

Memory Integrity Tree



Merkle hash tree for Integrity Verification of large amounts of data

Protecting Memory Confidentiality & Integrity – Processor chip is security perimeter

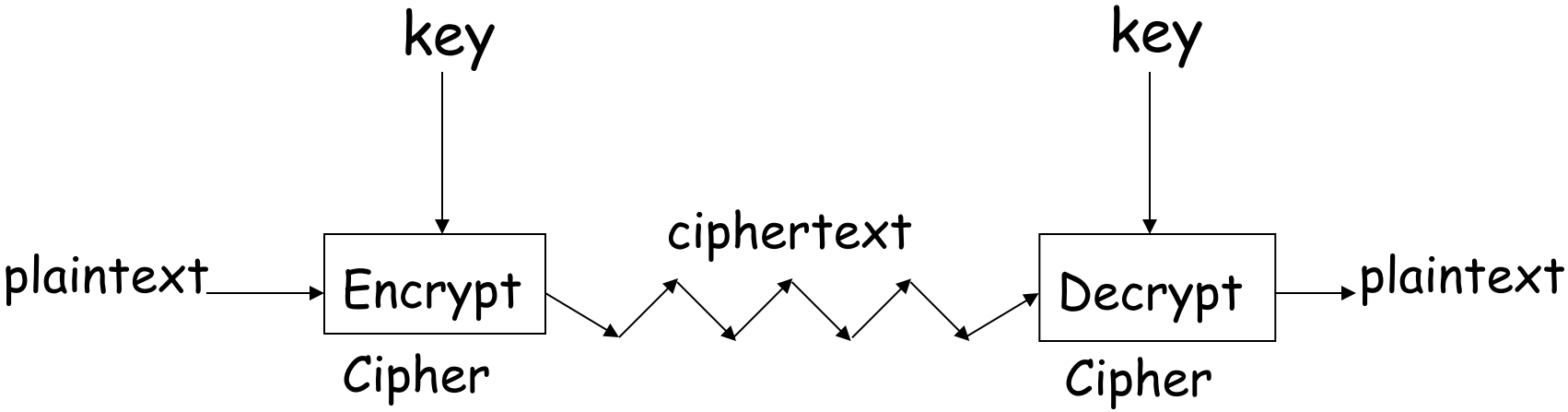


- Confidentiality: encrypt when evict cache-line from Last Level Cache (LLC), decrypt when fetch on-chip into LLC
- Integrity verification:
  - Check if a value from external memory is the most recent value stored at that address by the processor
  - Use cryptographic hash, MAC or Merkle hash tree

Importance of Key Management

- Kerckhoff's rule: crypto algorithm should be public, only the key must be secret
  - No "security by obscurity"
- Once key is known, any security provided by applying cryptography is lost
    - Secret key (symmetric key ciphers, and keyed-hashes)
    - Private key (public key ciphers)
  - Distinguish between Secret Key and Private Key
- Many keys needed
  - Keys can be generated when needed, or hierarchically encrypted (by SW or HW)
  - "master keys" should be protected by HW, since HW much less vulnerable to attacks

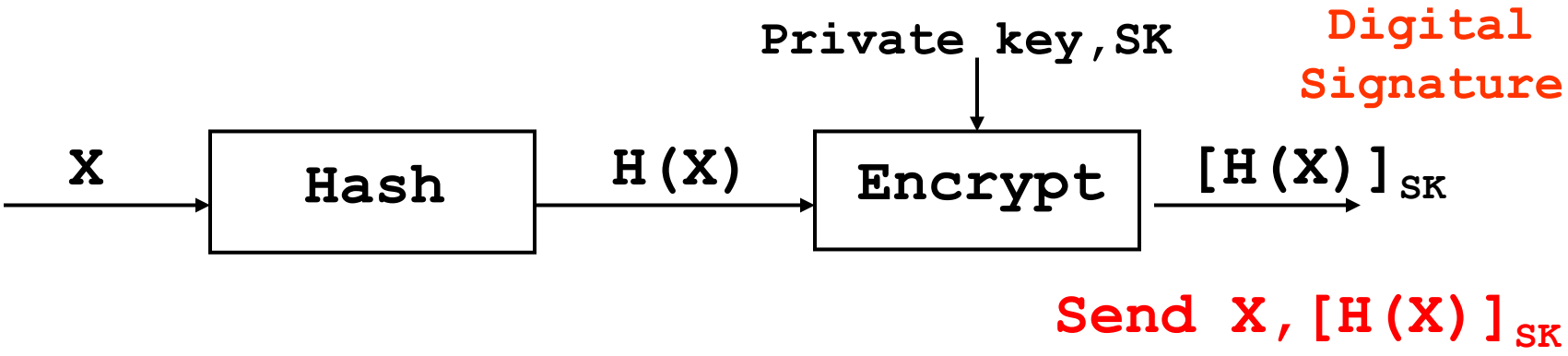
Public-Key Cryptography



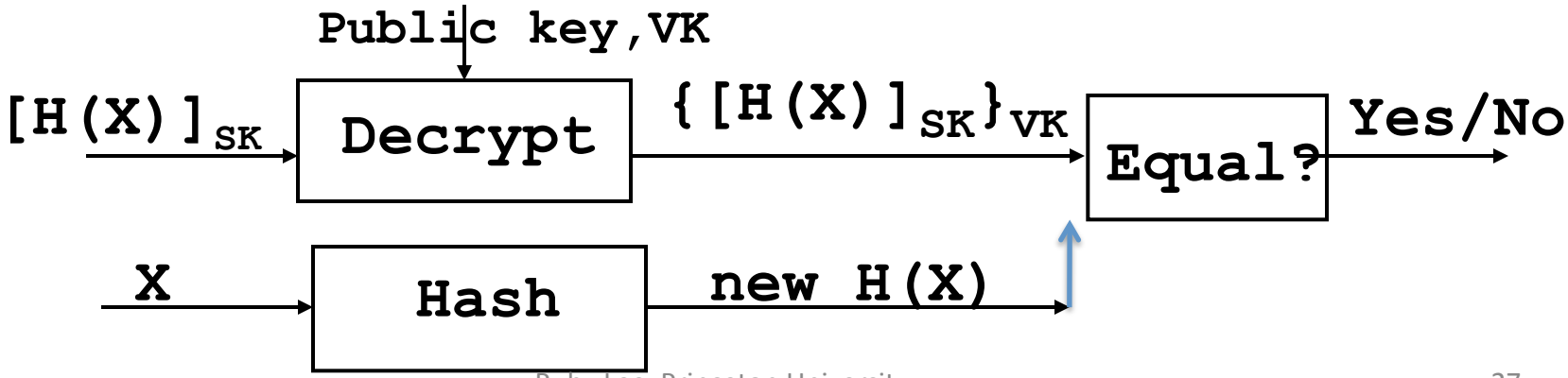
- Symmetric-key crypto
  - Shared secret key used for encryption and decryption
- Public-key (asymmetric-key) crypto:
  - Encryption with Public key of recipient, decryption with Private key of recipient
  - Signing with Private key of sender, Verification with Public key of sender

Digital Signature

Signature generation



Signature verification

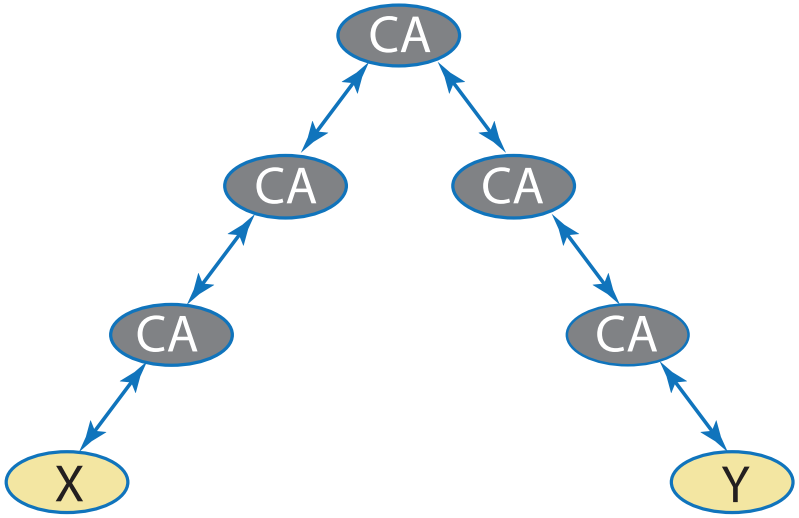


Public Key Infrastructure (PKI)

- How do you know a public key really belongs to Alice?
- Public keys have to be verified by a trusted authority, called a CA
- Certificate authority (CA) provides a way to obtain verified public-key certificates
  - CA verifies the credentials of a party before issuing a public-key certificate to it
- Public-key certificate has the following:
    - Name of certificate owner
    - Public-keys of the certificate owner
    - Period of validity for certificate
    - **Certificate has CA's digital signature**

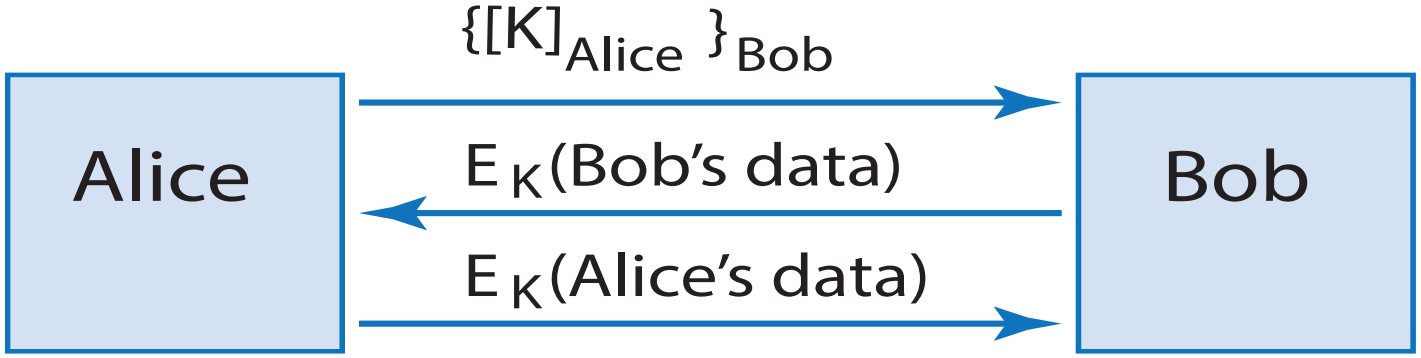
Chain of trust
(wrt to Public-Key Certificates)

- Chain of trust: a series of certificates, each signed by a CA at next higher level of trust hierarchy
  - e.g., Alice's certificate is signed by Princeton CA, whose certificate is signed by NJ CA, whose certificate is signed by Verisign USA CA.



Confidentiality using public and symmetric key crypto

- Use public-key crypto to establish symmetric key K, which is then used for bulk encryption (much faster)



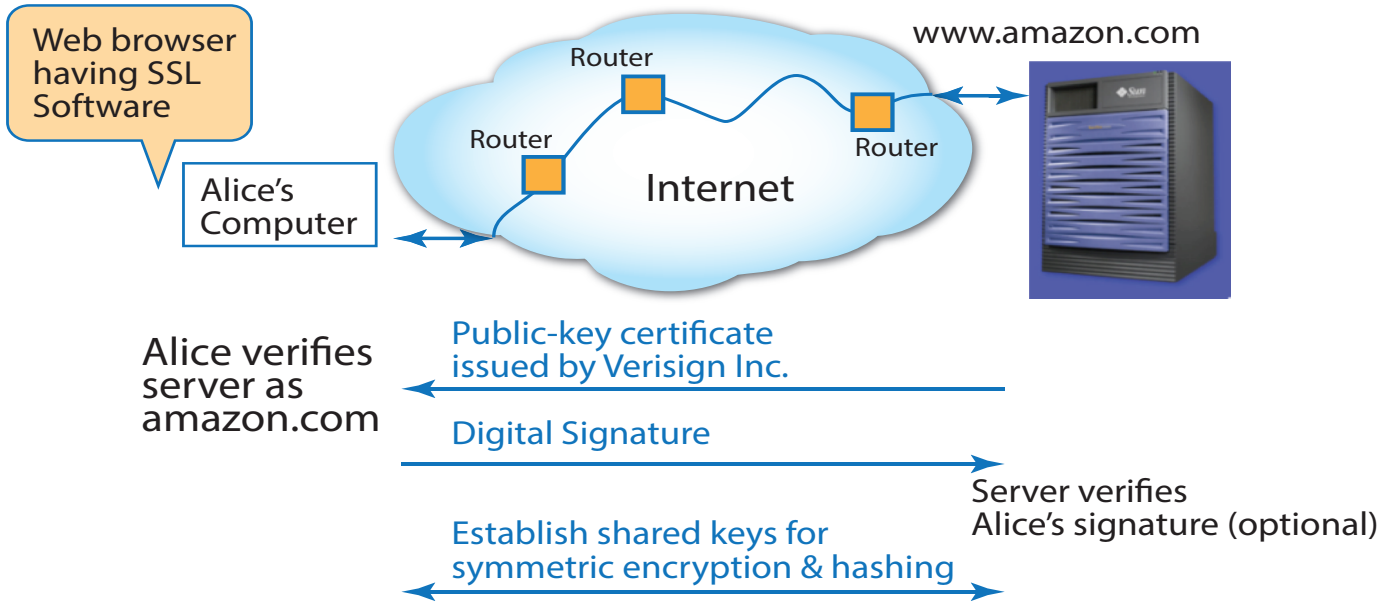
Protocol Notation:

Alice -> Bob: $\{[K]_{Alice}\}_{Bob}$; Alice sends Bob a secret key, signed & encrypted

Bob -> Alice: E_K(Bob's data) ; Bob uses this for confidential communications with Alice

Alice -> Bob: E_K(Alice's data) ; likewise for Alice with Bob.

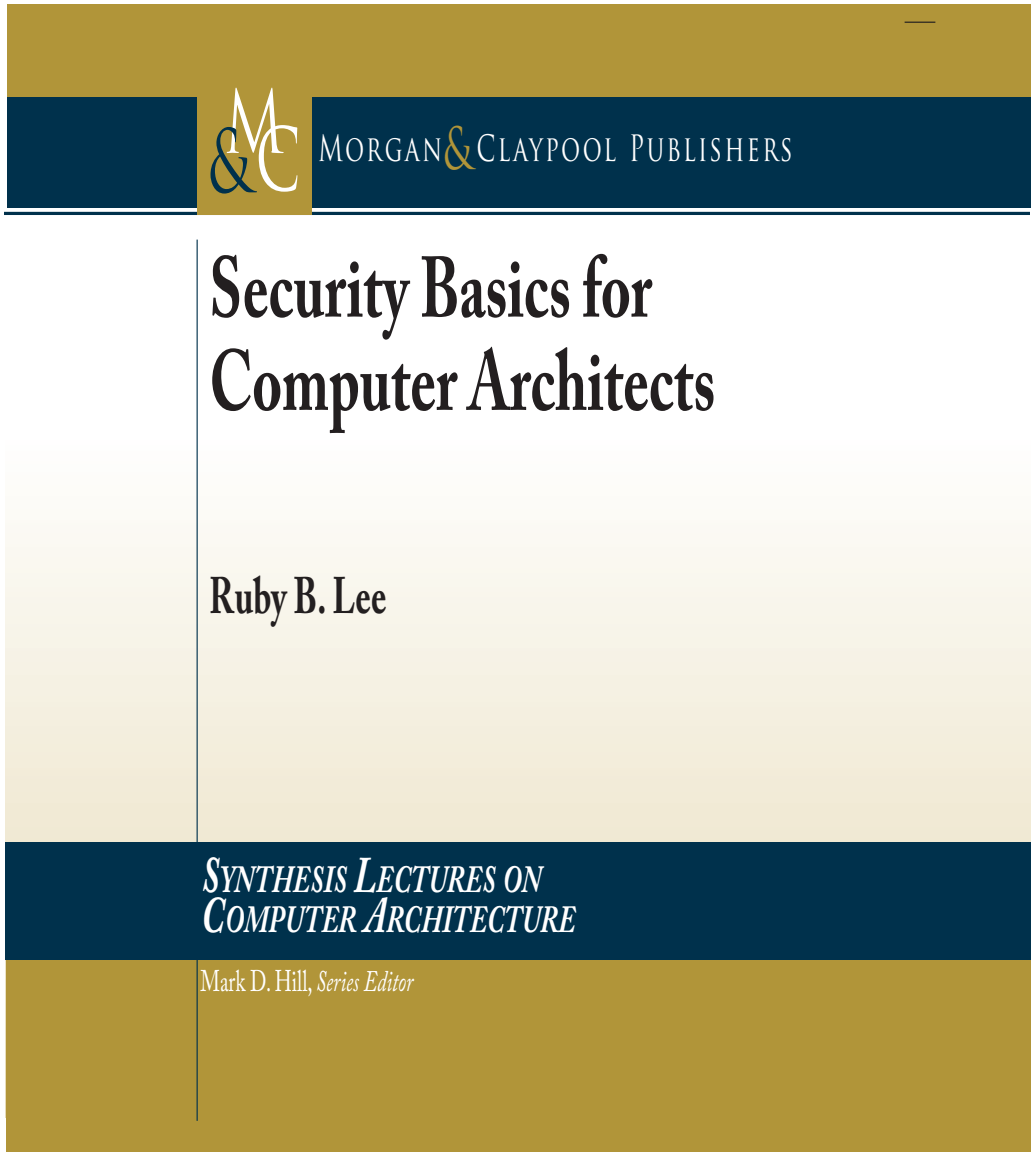
SSL (Secure Sockets Layer) protocol for securing web transactions



(a) **SSL Handshake**

(b) **SSL Record**

More Details and many References can be found in:



http://dx.doi.org/10.2200/S00512ED1V01Y201305CAC025