EE515
Think Like an Adversary

Yongdae Kim
KAIST

EE515
Security of Emerging Systems

Yongdae Kim
KAIST

SysSec
System Security Lab

Offense vs. Defense

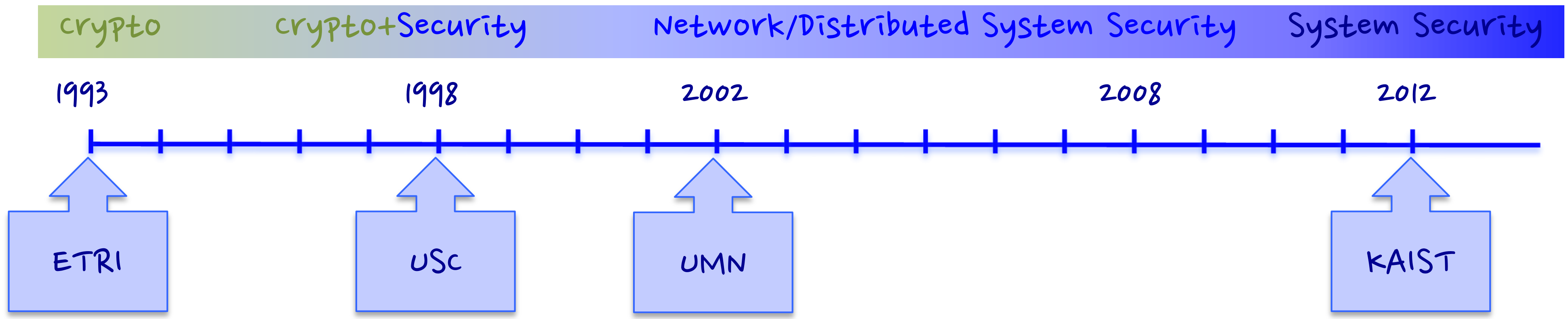
- “Know your enemy.” – Sun Tzu

- "the only real defense is active defense” - Mao Zedong

- “security involves **thinking like an attacker, an adversary or a criminal**. If you don’t see the world that way, you’ll never notice most security problems.” - Bruce Schneier

Instructor, TA, Office Hours

- Instructor
  - Yongdae Kim
    - 12th time teaching EE515/IS523
    - >42th time teaching a security class
  - Email: yongdaek (at) kaist. ac. Kr
    yongdaek (at) gmail. com
    - Please include ee515 in the subject of your mail
  - Office: N26 201
  - Office Hours: TBD
- TA
  - EE TA: Beomseok Oh, Sangmin Woo, Duckwoo Kim
  - GSIS TA: Hansung Bae, Kwangmin Kim
  - syssec.ta (at) gmail.com
  - Office hours: by appointment only



- 30+ year career in security research
  - Applied Cryptography, Group key agreement, Storage, P2P, Mobile/Sensor/Ad-hoc/Cellular Networks, Social networks, Internet, Anonymity, Censorship

- Published about ~150 papers (~11,000 Google scholar citations)

Class web page, e-mail

- http://security101.kr
  - Read the page carefully and regularly!
  - Read the Syllabus carefully.
  - Check calendar.

- E-mail policy
  - Include [ee515] in the subject of your e-mail

Textbook

- Required: Papers!

- Optional
  - Handbook of Applied Cryptography by Alfred J. Menezes, Paul C. Van Oorschot, Scott A. Vanstone (Editor), CRC Press, ISBN 0849385237, (October 16, 1996) Available on-line at http://www.cacr.math.uwaterloo.ca/hac/
  - Security Engineering by Ross Anderson, Available at http://www.cl.cam.ac.uk/~rja14/book.html.

Goals

- To discover new attacks in emerging systems
- The main objective of this course is to learn how to think like an adversary.
- Review various ingenuous attacks and discuss why and how such attacks were possible.
- Students who take this course will be able to analyze security of practical systems

No Goals

- In depth study of OS/Software/Network security and Cryptography
- Hands-on Hacking Tutorial on Android, Windows, Embedded Systems, etc.

Course Content

- Overview
  - Introduction
  - Attack Model, Security Economics, Legal Issues, Ethics
  - Cryptography and Key Management
- Frequent mistakes
  - User Interface and Psychological Failures
  - Software Engineering Failures and Malpractices
  - Cryptographic Failures
- Case Studies
  - Medical Device
  - Blockchain
  - Privacy
  - Machine Learning
  - Autonomous Driving
  - Drone
  - Cellular Network
  - Metaverse
  - Satellite?
  - Anything else?

Evaluation (IMPORTANT!)

- Approximately,
  - Lecture: 20 %
  - Reading report: 32.5 % (2.5 % x 13)
  - Project: 37.5 %
  - Participation: 10 %

Group Projects

- Each project should have some "research" aspect.
- Group size
  - Min 1 Max 5
- Important dates
  - Pre-proposal: 9/29/2024, 11:59 PM
  - Full Proposal: 10/13/2024, 11:59 PM
  - Midterm Report: 11/10/2024, 11:59 PM
  - Final Report: 12/18/2024, 11:59 PM
- Project examples
  - Attack, attack, attack!
  - Analysis
  - Measurement

Grading

- Absolute (i.e. not on a curve)
  - But flexible ;-)

- Grading will be as follows
  - 93.0% or above yields an A, 90.0% an A-
  - 85% = B+, 80% = B, 75% = B-
  - 70% = C+, 65% = C, 60% = C-
  - 55% = D+, 50% = D, and less than 50% yields an F.

Reading Report (Precise and Concise)

- Class day
  - Target System/Service: 5 pts
  - Vulnerability: 10 pts
  - Exploitation (Attacks): 10 pts
  - Evaluation and experimental method: 10 pts
  - Defense (potential solutions): 5 pts
  - Question to the presenter: 10 pts
- Next Class
  - Discussion
    - Discussion for question 1 - 10 points
    - Discussion for question 2 - 10 points
    - Discussion for question 3 - 10 points
    - Any supplemental discussion topics (future works, criticism, follow-up studies, …) - 20 points

And…

- Incompletes (or make up exams) will in general not be given.
  - Exception: a provably serious family or personal emergency arises with proof and the student has already completed all but a small portion of the work.

- Scholastic conduct must be acceptable. Specifically, you must do your assignments, quizzes and examinations yourself, on your own.

The Telegraph

HOME NEWS WORLD SPORT FINANCE COMMENT BLOGS CULTURE TRAVEL LIFE

Politics | Obits | Education | Earth | Science | Defence | Health | Scotland | Royal | Celebrities

HOME » NEWS » UK NEWS » CRIME

Thieves placed bugs and hacked onboard computers of luxury cars

The leader of a gang that hacked into the onboard computers of luxury cars and bugged them with GPS tracking devices before stealing them is facing jail.

Bloomberg Our Company | Professional | Anywhere

HOME QUICK NEWS OPINION MARKET DATA PERSONAL FINANCE TECH POLITICS

McAfee Hacker Says Medtronic Insulin Pumps Vulnerable To Attack

Confirmed: US and Israel created Stuxnet, lost control of it

Stuxnet was never meant to propagate in the wild.

by Nate Anderson - June 1 2012, 6:00am EDT

HACKING NATIONAL SECURITY 277

KrebsonSecurity

In-depth security news and investigation

FBI: Smart Meter Hacks Likely to Spread

Iran's Flying Saucer Downed U.S. Drone, Engineer Claims

By Spencer Ackerman and Noah Shachtman | January 10, 2012 | 1:00 pm |

Categories: Tinfoil Tuesday

Most CCTV systems are easily accessible to attackers

Andy Greenberg, Forbes Staff

Covering the worlds of data security, privacy and hacker culture.

+ Follow (512)

SPONSORED BY sas

SECURITY | 7/23/2012 @ 12:17PM | 218,082 views

Hacker Will Expose Potential Security Flaw In Four Million Hotel Room Keycard Locks

The cyberweapon that could take down the internet

› 13:30 11 February 2011 by Jacob Aron

› For similar stories, visit the Computer crime Topic Guide

27th Chaos Communication Congress

We come in peace

Wideband GSM Sniffing

The Telegraph

HOME NEWS WORLD SPORT FINANCE COMMENT BLOGS CULTURE TRAVEL LIFE FASHI

Politics | Obits | Education | Earth | Science | Defence | Health | Scotland | Royal | Celebrities

HOME » NEWS

Marie Colvin: Syria regime accused of murder in besieged Homs

Cyber War and Ukraine

3 reasons Moscow isn't taking down Ukraine's cell networks

Russian Army Using Simboxes on Ukrainian Networks

Chinese drone firm DJI pauses operations in Russia and Ukraine

DJI ADMITS DRONE AEROSCOPE SIGNALS ARE NOT ACTUALLY ENCRYPTED

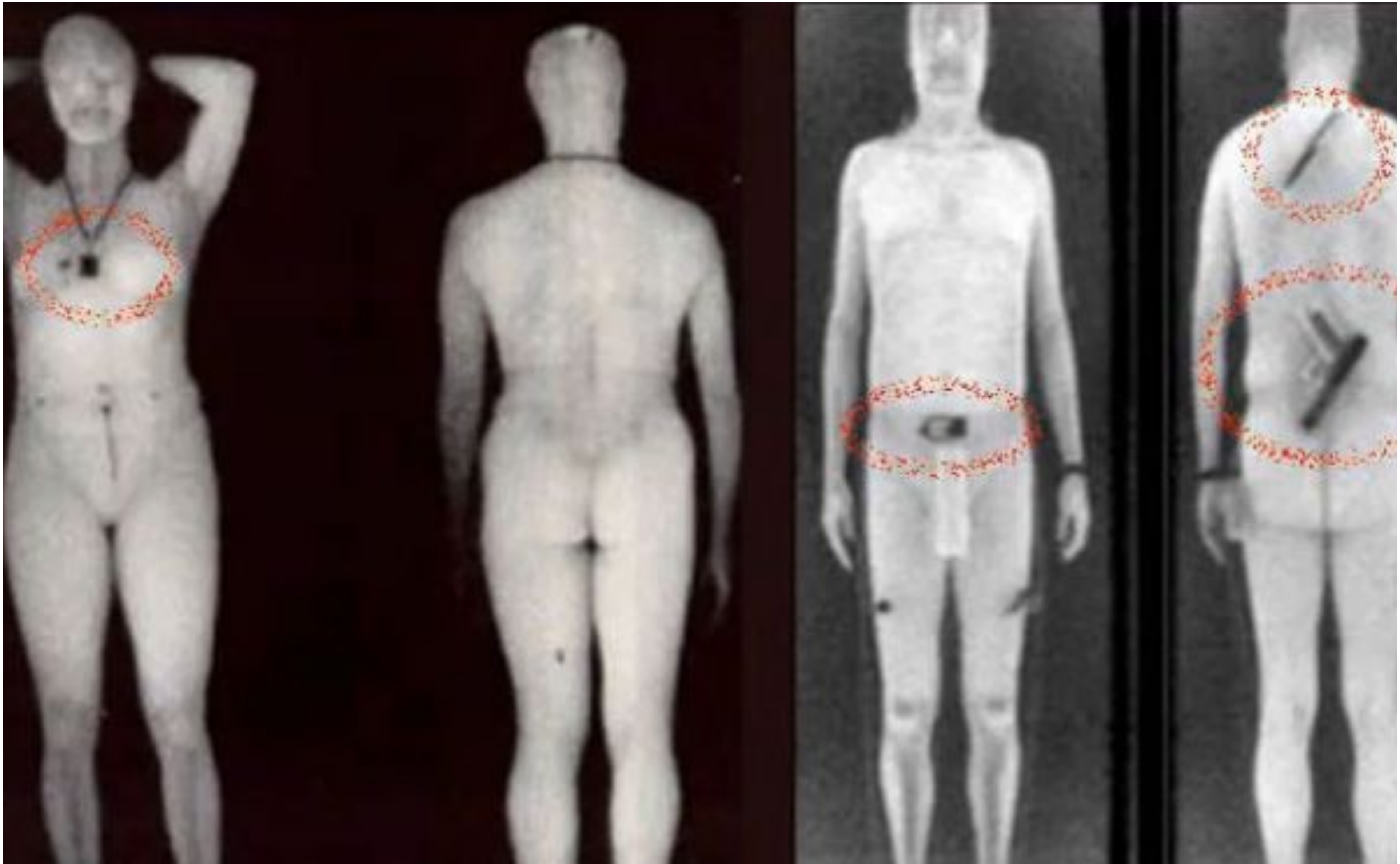
UkraineX: How Elon Musk's space satellites changed the war on the ground

Security Engineering

- Building a systems to remain dependable in the face of malice, error or mischance

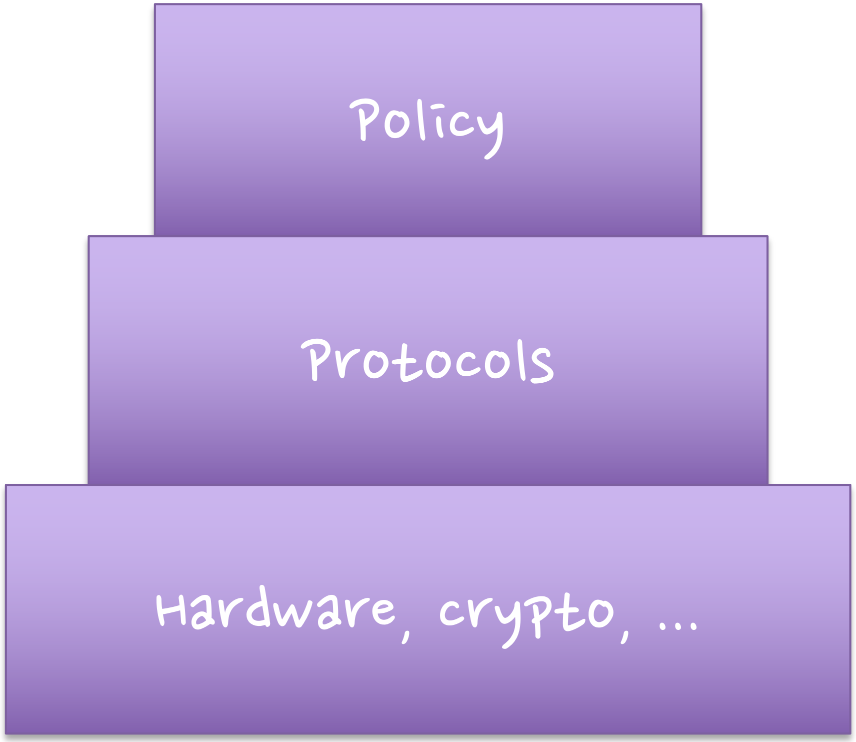
System	Service	Attack Deny Service, Degrade QoS, Misuse	Security Prevent Attacks
Communication	Send message	Eavesdrop	Encryption
Web server	Serving web page	DoS	CDN?
Computer	;-)	Botnet	Destroy
SMS	Send SMS	Shutdown Cellular Network	Rate Control, Channel separation
Pacemaker	Heartbeat Control	Remote programming and eavesdropping	Distance bounding?
Nike+iPod	Music + Pedometer	Tracking	Don't use it?
Recommendation system	Collaborative filtering	Control rating using Ballot stuffing	?

TSA Body Scanner



Design Hierarchy

- What are we trying to do?
- How?
- With what?
- Considerations
  - Top-down vs. Bottom-up
  - Iterative
  - Convergence
  - environment change



Goals: Confidentiality

- Confidentiality of information means that it is accessible only by authorized entities
  - Contents, Existence, Availability, Origin, Destination, Ownership, Timing, etc… of:
  - Memory, processing, files, packets, devices, fields, programs, instructions, strings...

Goals: Integrity

- Integrity means that information can only be modified by authorized entities
  - e.g. Contents, Existence, Availability, Origin, Destination, Ownership, Timing, etc… of:
  - Memory, processing, files, packets, devices, fields, programs, instructions, strings...

Goals: Availability

- Availability means that authorized entities can access a system or service.

- A failure of availability is often called Denial of Service:
  - Packet dropping
  - Account freezing
  - Jamming
  - Queue filling

Goals: Accountability

- Every action can be traced to "the responsible party."

- Example attacks:
  - Microsoft cert
  - Guest account
  - Stepping stones

Goals: Dependability

- A system can be relied on to correctly deliver service
- Dependability failures:
  - Therac-25: a radiation therapy machine
    - » whose patients were given massive overdoses (100 times) of radiation
    - » bad software design and development practices: impossible to test it in a clean automated way
  - Ariane 5: expendable launch system
    - » the rocket self-destructing 37 seconds after launch because of a malfunction in the control software
    - » A data conversion from 64-bit floating point value to 16-bit signed integer value

Interacting Goals

- Failures of one kind can lead to failures of another, e.g.:
  - Integrity failure can cause Confidentiality failure
  - Availability failure can cause integrity, confidentiality failure
  - Etc…

Threat Model

- What property do we want to ensure against what adversary?

- Who is the adversary?
- What is his goal?
- What are his resources?
  - e.g. Computational, Physical, Monetary…
- What is his motive?
- What attacks are out of scope?

Terminologies

- Attack (Exploit): attempt to breach system security (DDoS)
- Threat: a scenario that can harm a system (System unavailable)
- Vulnerability: the “hole” that allows an attack to succeed (TCP)
- Security goal: “claimed” objective; failure implies insecurity

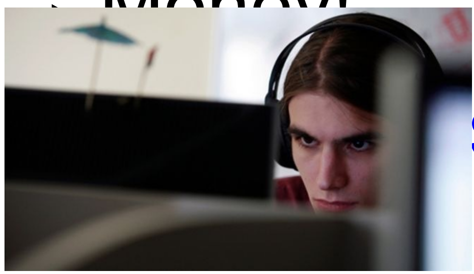
Who are the attackers?

- No more script-kiddies
- sore
  a n
-

  mput npute
  f prote te po
- Orga inal
  - Money!
- s





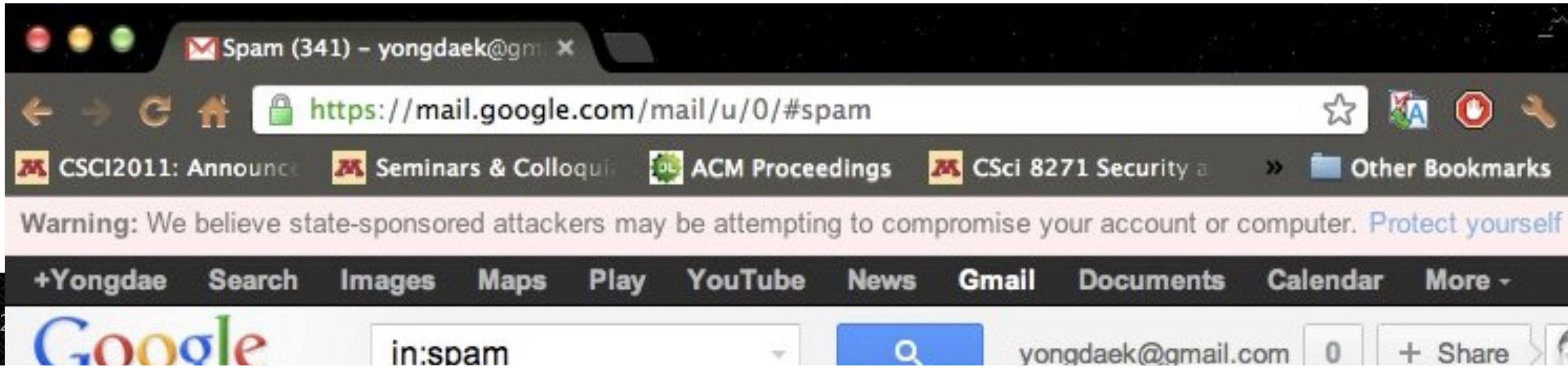




State-Sponsored Attackers

- 2012. 6: Google starts warning users who may be targets of government-sponsored hackers

- 2010 ~: Stuxnet, Duqu, Flame, Gauss, …
  - Mikko (2011. 6): A Pandora's Box We Will Regret Opening

- 2010 ~: Cyber Espionage from China
  - Exxon, Shell, BP, Marathon Oil, ConocoPhillips, Baker Hughes
  - Canada/France Commerce Department, EU parliament
  - RSA Security Inc. SecurID
  - Lockheed Martin, Northrop Grumman, Mitsubushi



Hacktivists

- promoting expressive politics, free speech, human rights, and information ethics
- Anonymous
  - To protest against SOPA, DDoS against MPAA, RIAA, FBI, DoJ, Universal music
  - Attack Church of Scientology
  - Support Occupy Wall Street

- LulzSec
  - Hacking Sony Pictures (PSP jailbreaking)
  - Hacking Pornography web sites
  - DDoSing CIA web site (3 hour shutdown)

Security Researchers

- They tried to save the world by introducing new attacks on systems

- Examples
  - Diebold AccuVote-TS Voting Machine
  - APCO Project 25 Two-Way Radio System
  - Kad Network
  - GSM network
  - Pacemakers and Implantable Cardiac Defibrillators
  - Automobiles, …

Rules of Thumb

- Be conservative: evaluate security under the best conditions for the adversary
- A system is as secure as the weakest link.
- It is best to plan for unknown attacks.

Security & Risk

- The risk due to a set of attacks is the expected (or average) cost per unit of time.
- One measure of risk is Annualized Loss Expectancy, or ALE:

$$\sum_{\text{attack } A} (p_A \times L_A)$$

ALE of attack A: $(p_A \times L_A)$

p_A: Annualized attack incidence

L_A: Cost per attack

Risk Reduction

- A defense mechanism may reduce the risk of a set of attacks by reducing L_A or p_A. This is the gross risk reduction (GRR):

$$\sum_{\text{attack } A} (p_A \times L_A - p'_A \times L'_A)$$

- The mechanism also has a cost. The net risk reduction (NRR) is GRR – cost.

Bug Bounty Program

- Evans (Google): “Seeing a fairly sustained drop-off for the Chromium”
- McGeehan (Facebook): The bounty program has actually outperformed the consultants they hire.
- Google: Patching serious or critical bugs within 60 days
- Google, Facebook, Microsoft, Mozilla, Samsung, …

Nations as a Bug Buyer

- ReVuln, Vupen, Netragard: Earning money by selling bugs
- "All over the world, from South Africa to South Korea, business is booming in what hackers call zero days"
- "No more free bugs."
- 'In order to best protect my country, I need to find vulnerabilities in other countries'
- Examples
  - Critical MS Windows bug: $150,000
  - a zero-day in iOS system sold for $500,000
  - Vupen charges $100,000/year for catalog and bug is sold separately
  - Brokers get 15%.

Sony vs. Hackers

2000.8	2005.10	2007.1	2011.1	2011.4	2011.4
Sony Exec	Russinovich	FTC	Hotz	Sony, Hotz	PSN
do whatever to protect re	Sony	Reimburse	PS3 Hack	settled	Hacked

2011.

Son

½ day

recov

2011. 3 $36.27 per share

2011. 6 $24.97 per share

2011.5

ny Exec

ologized

2011.5	2011.5	2011.6	2012.3
SOE	Sony	Sony	Anon
Hacked	outage cost $171M	Fired security staff	Posted Unreleased Michael Jackson video

Patco Construction vs. Ocean Bank

- Hacker stole ~$600K from Patco through Zeus
- The transfer alarmed the bank, but ignored

- “commercially unreasonable”
  - Out-of-Band Authentication
  - User-Selected Picture
  - Tokens
  - Monitoring of Risk-Scoring Reports

Auction vs. Customers

- Auction's fault
  - Unencrypted Personal Information
  - It did not know about the hacking for two days
  - Passwords
    - » 'auction62', 'auctionuser', 'auction'
  - Malwares and Trojan horse are found in the server.

- Not gulity, because
  - Hacker utilized new technology, and were well-organized.
  - Auctions have too many server.
  - AVs have false alarms.
  - For large company like auction, difficult to use.
  - Causes massive traffic.

Cost of Data Breach

Ponemon Cost of Data Breach Study: 12th year in measuring cost of data breach

Company	Year	Data	Cost (USD)
Anthem	2015	80 M patient and employee records	100M
Ashley Madison	2015	33 M user accounts	850M
Ebay	2014	145M customer accounts	200M
JPMorgan Chase	2014	Financial/Personal Info of 76 M Personal, 7M Small B	1000M
Home Depot	2014	56 M credit card and 53 M email addresses.	80 M
Sony Pictures	2014	Personal Information of 3,000 employees	35 M
Target	2013	40 M credit and debit card, 70 M customer	252 M
Global Payments	2012	1.5M card accounts	90 M
Tricare	2011	5 M Tricare Military Beneficiary	130 M
Citi Bank	2011	360,000 Credit Card	19 M
Hearland	2009	130M Card	2800 M

Security theater is the practice of

- investing in countermeasures intended to provide the feeling of improved security
- while doing little or nothing to actually achieve it

- Bruce Schneier

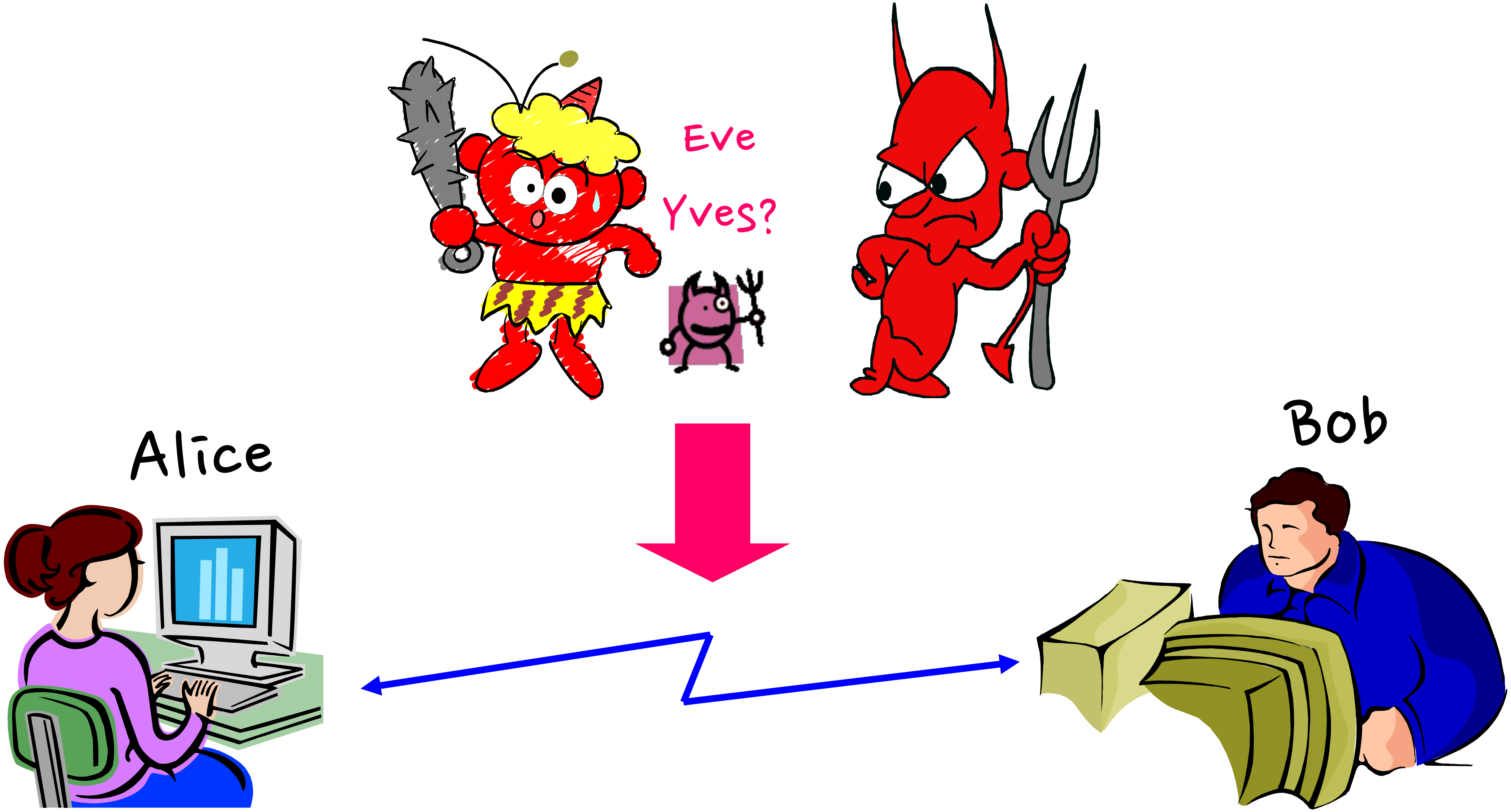
Security of New Technologies

- Most of the new technologies come with new and old vulnerabilities.
  - Old vulnerabilities: OS, Network, Software Security, …
  - Studying old vulnerabilities is important, yet less interesting.
  - e.g. Stealing Bitcoin wallet, Drone telematics channel snooping

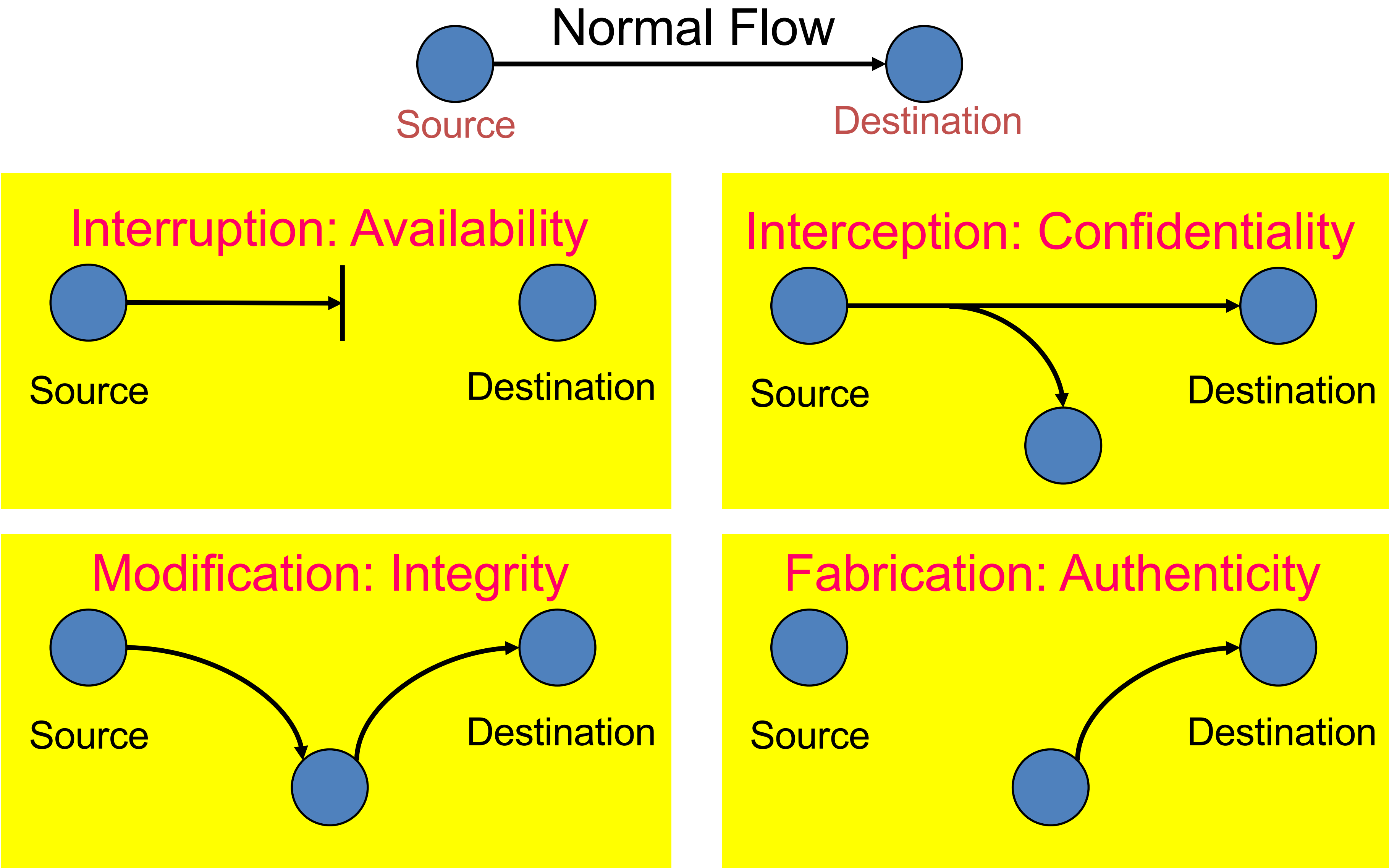
- New Problems in New Technologies
  - Sensors in Self-Driving Cars and Drones
  - Security of Deep Learning
  - Block Chain Pool Mining Attacks
  - Brain Hacking

Basic Cryptography

The Main Players



Attacks



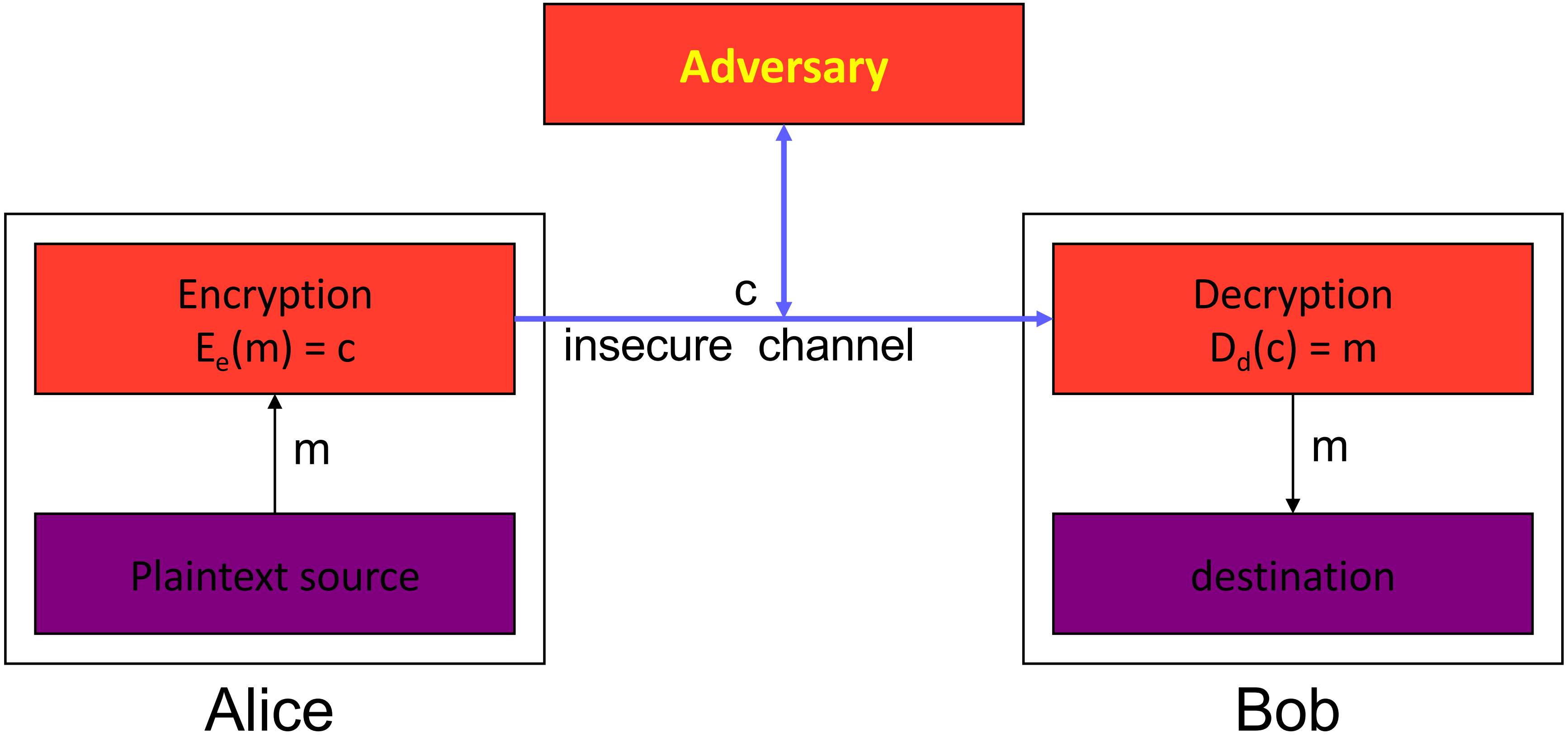
Normal Flow
Source
Destination
Interruption: Availability
Source
Destination
Interception: Confidentiality
Source
Destination
Modification: Integrity
Source
Destination
Fabrication: Authenticity
Source
Destination

Taxonomy of Attacks

- Passive attacks
  - Eavesdropping
  - Traffic analysis

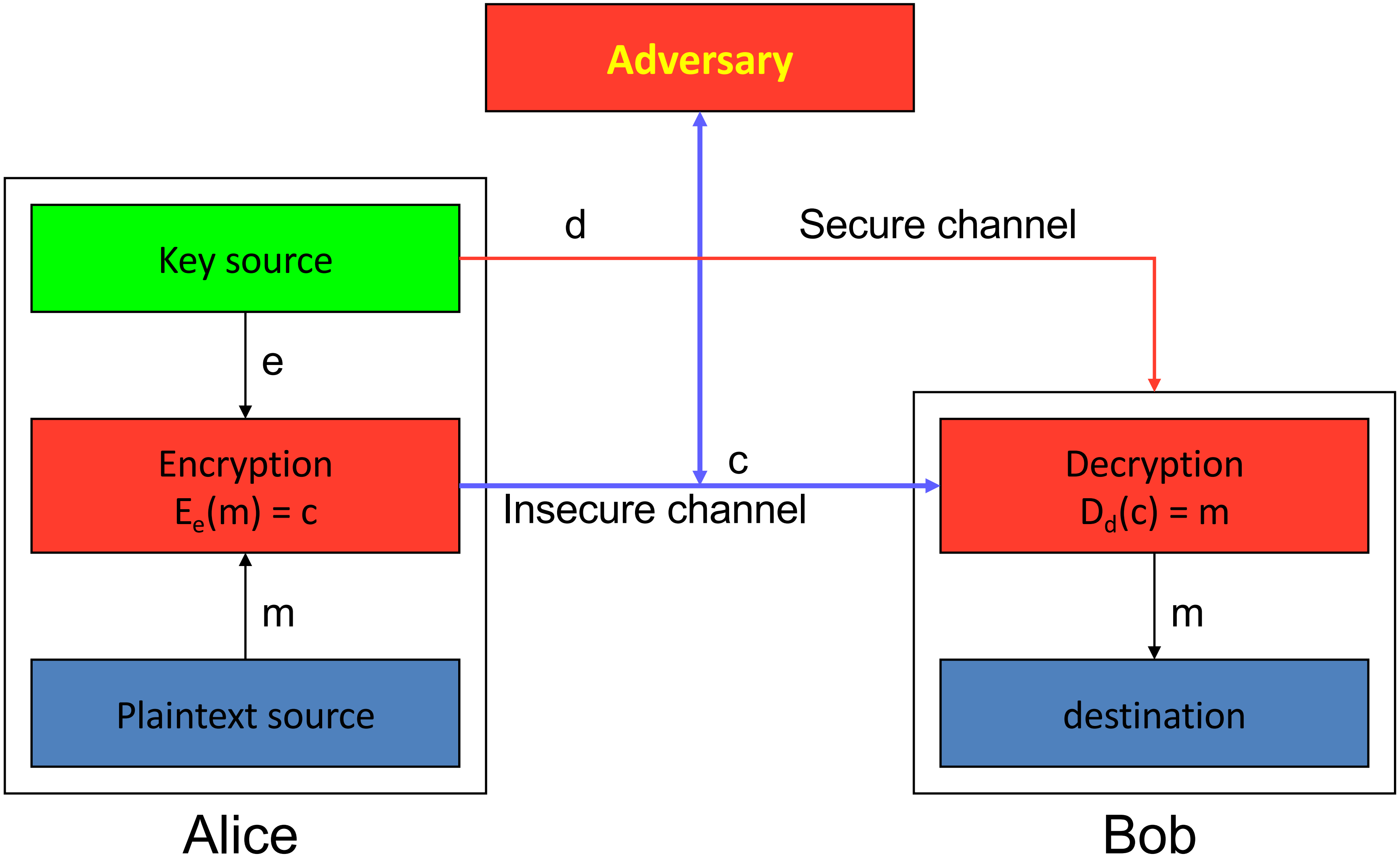
- Active attacks
  - Masquerade
  - Replay
  - Modification of message content
  - Denial of service

Encryption

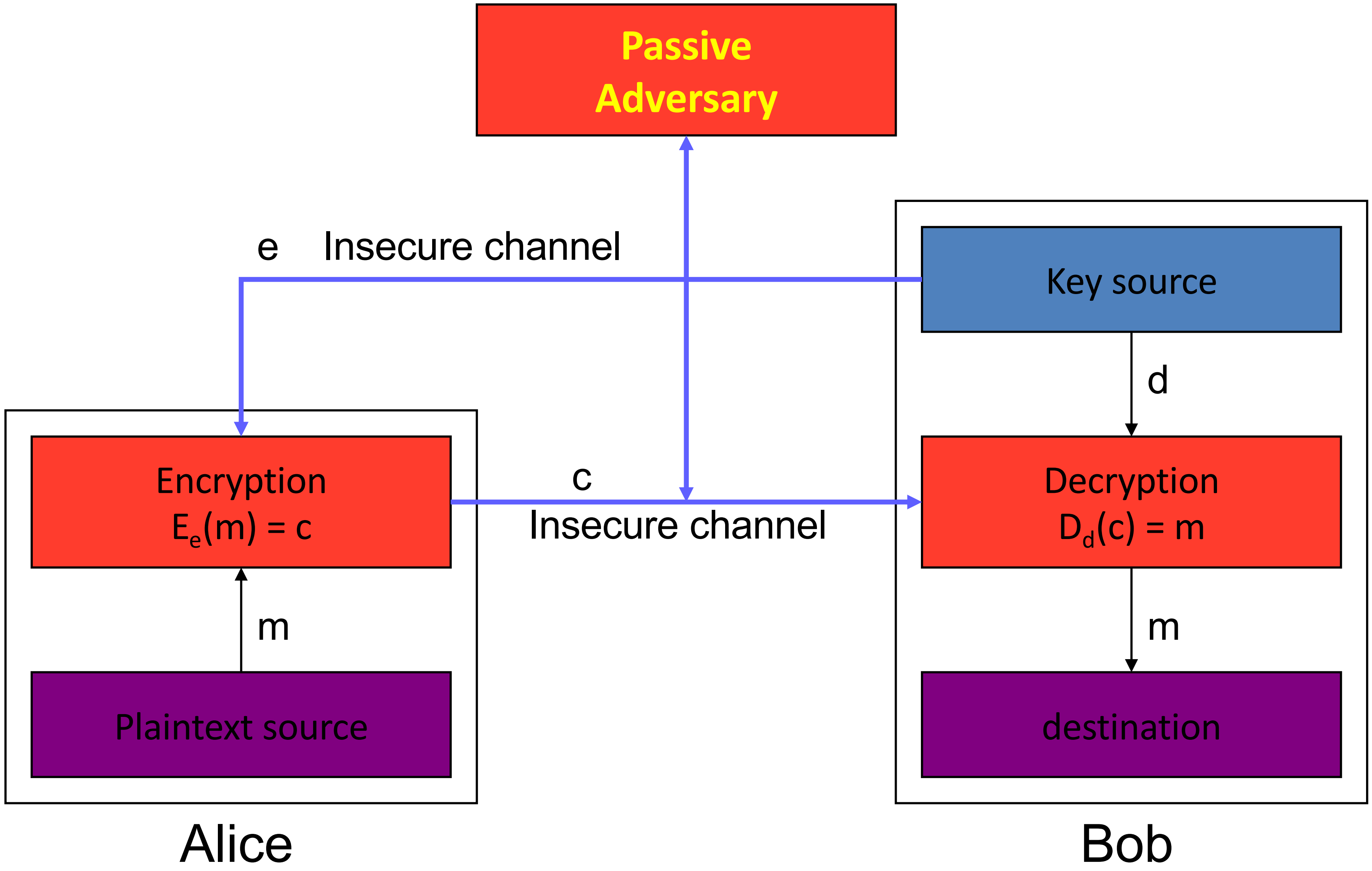


- Why do we use key?
  - Or why not use just a shared encryption function?

SKE with Secure channel

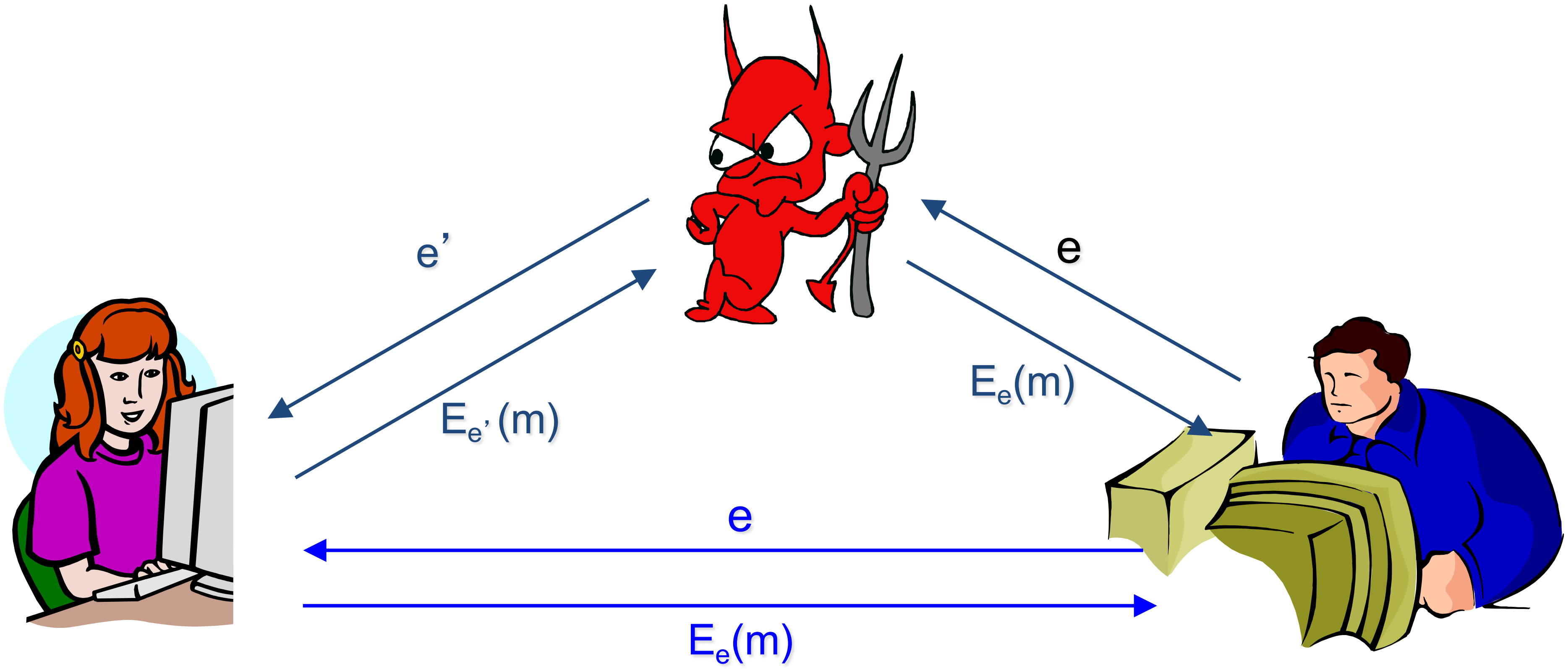


PKE with Insecure Channel



SysSec
System Security Lab

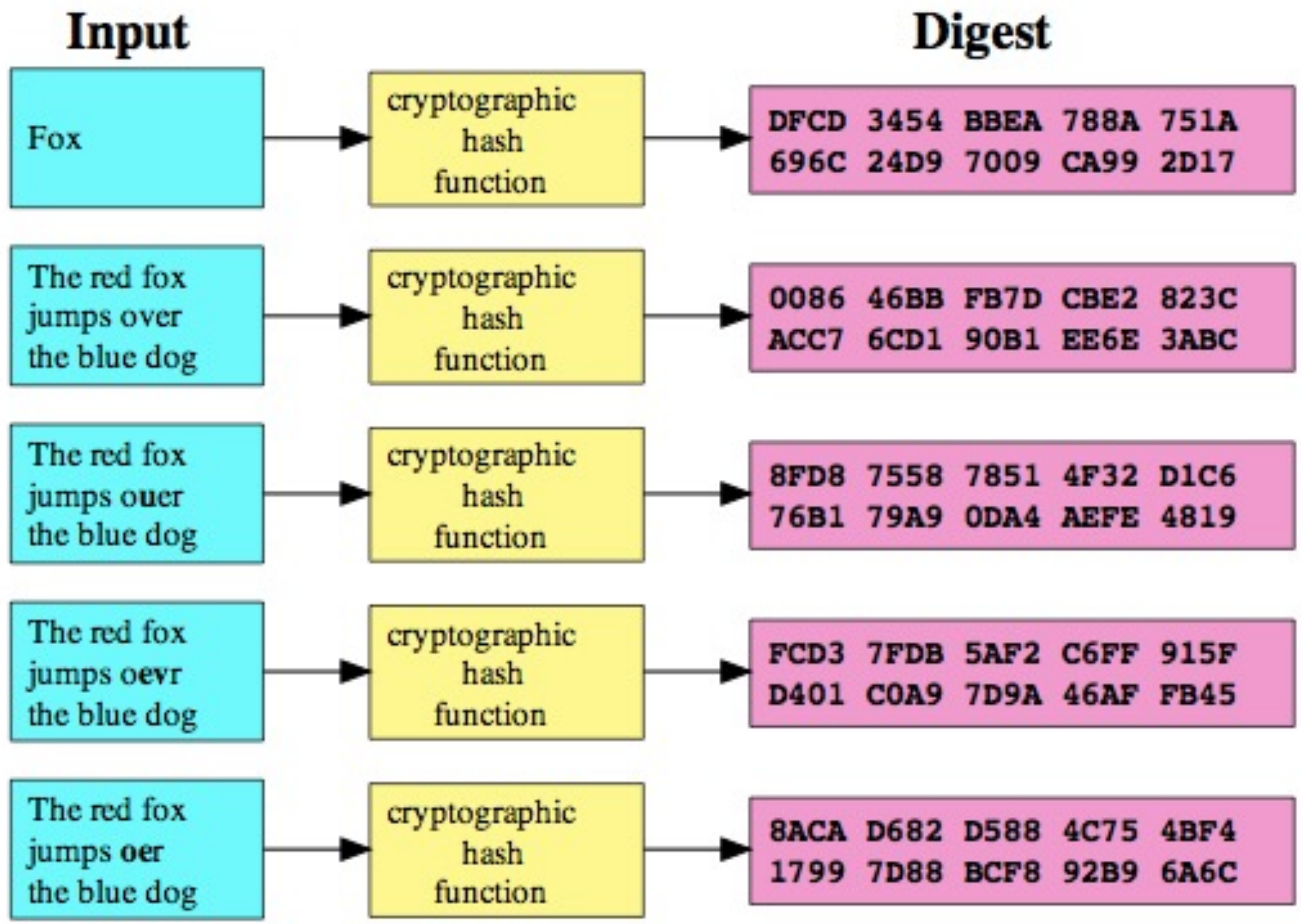
Public Key should be authentic!



Hash Function

- A hash function is a function h satisfying
  - $h:\{0, 1\}^* \rightarrow \{0, 1\}^k$ (Compression)
- A cryptographic hash function is a hash function satisfying
  - It is easy to compute y=h(x) (ease of computation)
  - For a given y, it is hard to find x' such that h(x')=y. (onewayness)
  - It is hard to find x and x' such that h(x)=h(x') (collision resistance)
- Examples: SHA-1, MD-5

How Random is the Hash function?



Applications of Hash Function

- File integrity


Instructions

The Windows SDK is available as a DVD ISO image file so that you can bu
that you are downloading the correct ISO file, please refer to the table bel
to validate that the file you've downloaded is the correct file.

==

File Name: GRMSDK_EN_DVD.iso

Chip: X86

CRC#: 0xCA4FE79D WalkerNews.net

SHA1: 0x8695F5E6810D84153181695DA78850988A923F4E

- Digital signature

  Sign = $S_{SK}(h(m))$

- Password verification

stored hash = h(password)

- File identifier
- Hash table
- Generating random numbers

Hash function and MAC

- A hash function is a function h
  - compression
  - ease of computation
  - Properties
    - » one-way: for a given y, find x' such that h(x') = y
    - » collision resistance: find x and x' such that h(x) = h(x')
  - Examples: SHA-1, MD-5

- MAC (message authentication codes)
  - both authentication and integrity
  - MAC is a family of functions h_k
    - » ease of computation (if k is known !!)
    - » compression, x is of arbitrary length, $h_k(x)$ has fixed length
    - » computation resistance
  - Example: HMAC

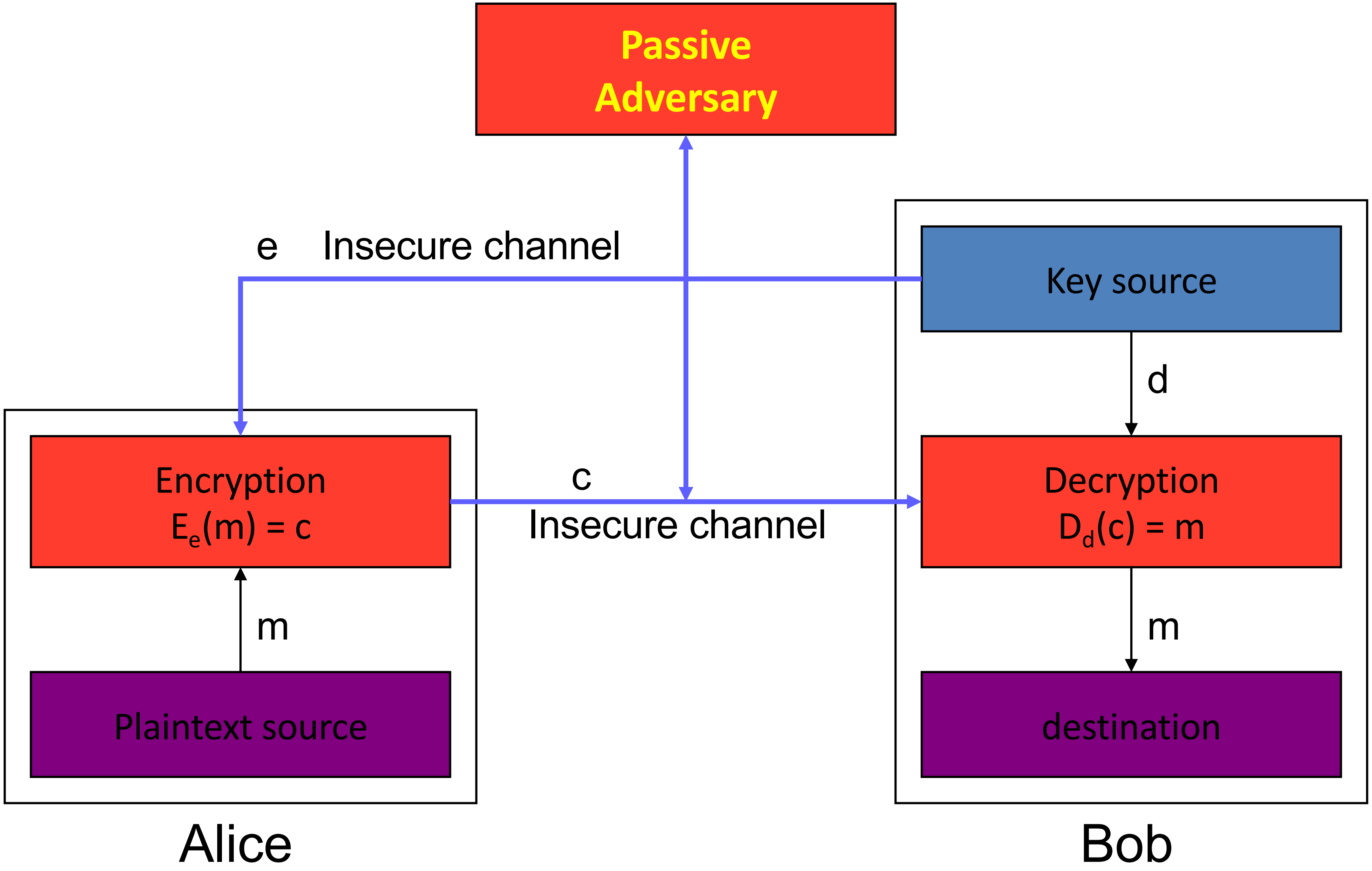
MAC construction from Hash

- Prefix
  - M=h(k||x)
  - appending y and deducing h(k||x||y) form h(k||x) without knowing k
- Suffix
  - M=h(x||k)
  - possible a birthday attack, an adversary that can choose x can construct x' for which h(x)=h(x') in $O(2^{n/2})$
- STATE OF THE ART: HMAC (RFC 2104)
  - HMAC(x)=h(k||p_1||h(k|| p_2||x)), p1 and p2 are padding
  - The outer hash operates on an input of two blocks
  - Provably secure

How to use MAC?

- A & B share a secret key k
- A sends the message x and the MAC $M \leftarrow H_k(x)$
- B receives x and M from A
- B computes $H_k(x)$ with received M
- B checks if $M = H_k(x)$

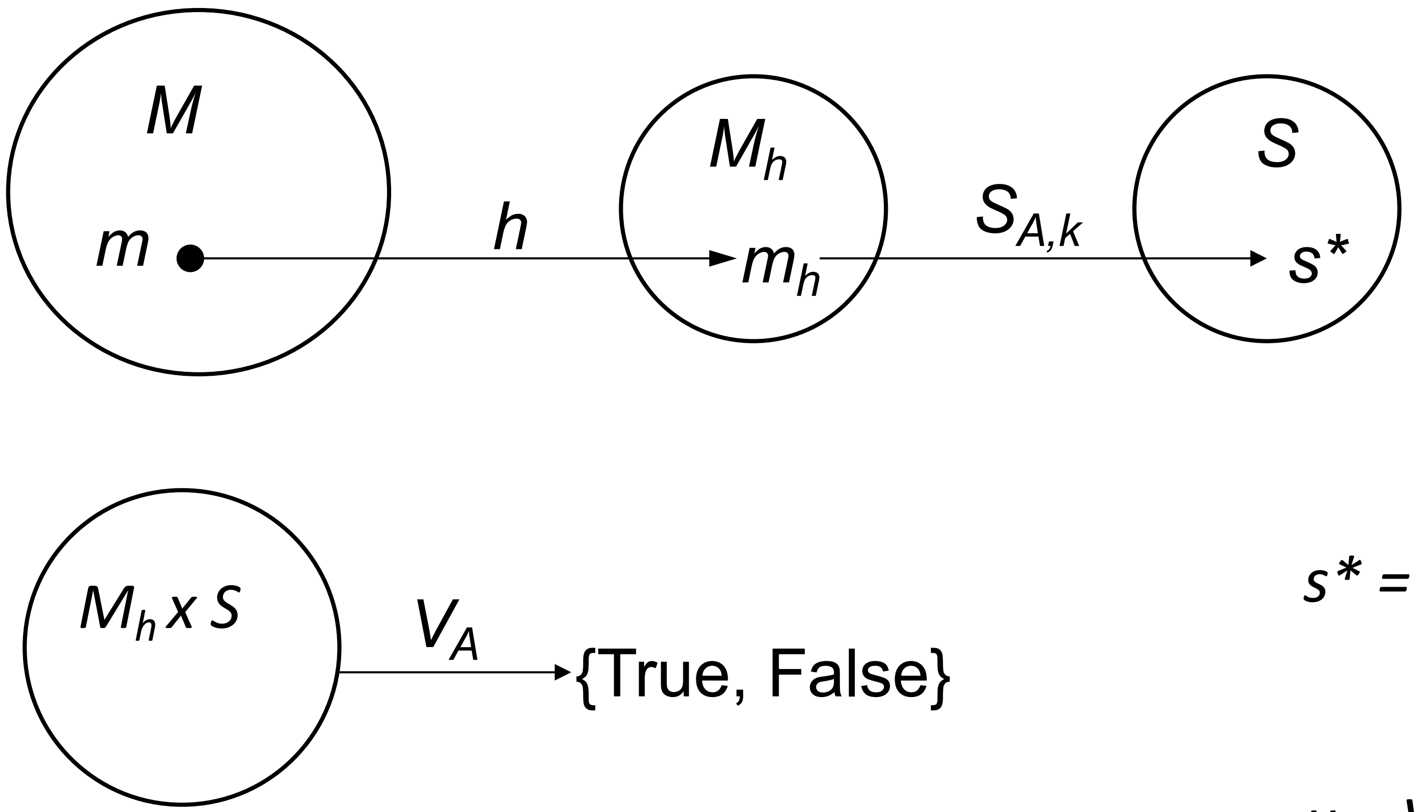
PKE with Insecure Channel



Digital Signature

- Integrity
- Authentication
- Non-repudiation

Digital Signature with Appendix



$s^* = S_{A,k}(m_h)$

$u = V_A(m_h, s^*)$

Authentication

- How to prove your identity?
  - Prove that you know a secret information
- When key K is shared between A and Server
  - A ➔ S: $HMAC_K(M)$ where M can provide freshness
  - Why freshness?
- Digital signature?
  - A ➔ S: $Sig_{SK}(M)$ where M can provide freshness
- Comparison?

Encryption and Authentication

- $E_K(M)$

- Redundancy-then-Encrypt: $E_K(M, R(M))$
- Hash-then-Encrypt: $E_K(M, h(M))$
- Hash and Encrypt: $E_K(M), h(M)$
- MAC and Encrypt: $E_{h1(K)}(M), HMAC_{h2(K)}(M)$
- MAC-then-Encrypt: $E_{h1(K)}(M, HMAC_{h2(K)}(M))$

Challenge-response authentication

- Alice is identified by a *secret* she possesses
  - *Bob* needs to know that Alice does indeed possess this secret
  - *Alice* provides ***response*** to a time-variant ***challenge***
  - Response depends on ***both*** secret and challenge

- Using
  - Symmetric encryption
  - One way functions

Challenge Response using SKE

- Alice and Bob share a key *K*
- Taxonomy
  - ***Unidirectional*** authentication using ***timestamps***
  - ***Unidirectional*** authentication using ***random numbers***
  - ***Mutual*** authentication using ***random numbers***
- Unilateral authentication using timestamps
  - Alice $\rightarrow$ Bob: $E_K(t_A, B)$
  - Bob decrypts and verified that timestamp is OK
  - Parameter *B* prevents replay of same message in B $\rightarrow$ A direction

Challenge Response using SKE

- Unilateral authentication using random numbers
  - Bob → Alice: r_b
  - Alice → Bob: $E_K(r_b, B)$
  - Bob checks to see if r_b is the one it sent out
    - » Also checks "B" – prevents reflection attack
  - r_b must be ***non-repeating***
- Mutual authentication using random numbers
  - Bob → Alice: r_b
  - Alice → Bob: $E_K(r_a, r_b, B)$
  - Bob → Alice: $E_K(r_a, r_b)$
  - Alice checks that r_a, r_b are the ones used earlier

Challenge-response using OWF

- Instead of encryption, used keyed MAC h_K
- Check: compute MAC from *known quantities,* and check with message
- SKID3
  - Bob $\rightarrow$ Alice: r_b
  - Alice $\rightarrow$ Bob: r_a, $h_K(r_a, r_b, B)$
  - Bob $\rightarrow$ Alice: $h_K(r_a, r_b, A)$

Key Establishment, Management

- Key establishment
  - Process to whereby a shared secret key becomes available to two or more parties
  - Subdivided into key agreement and key transport.

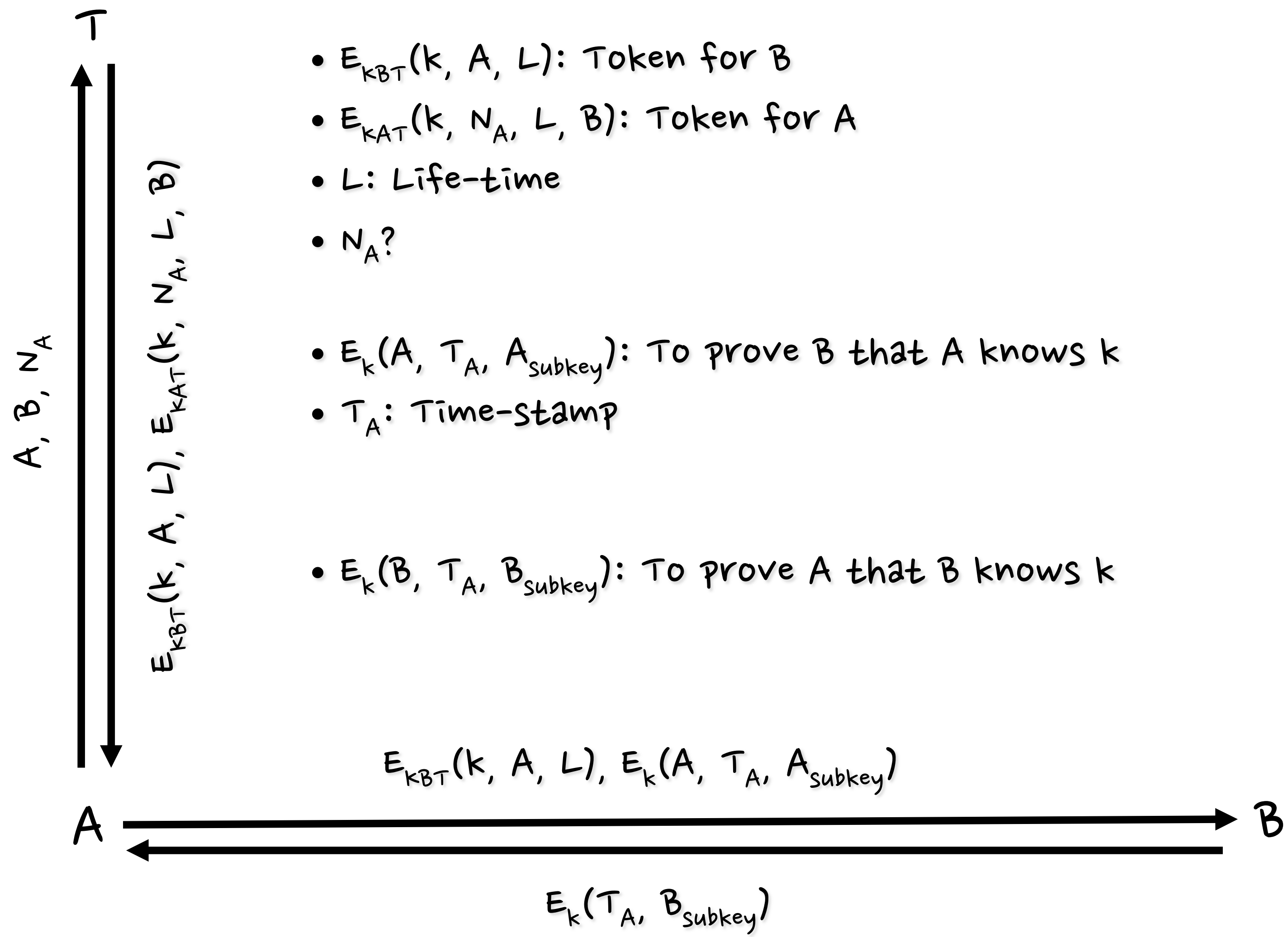
- Key management
  - The set of processes and mechanisms which support key establishment
  - The maintenance of ongoing keying relationships between parties

Kerberos vs. PKI vs. IBE

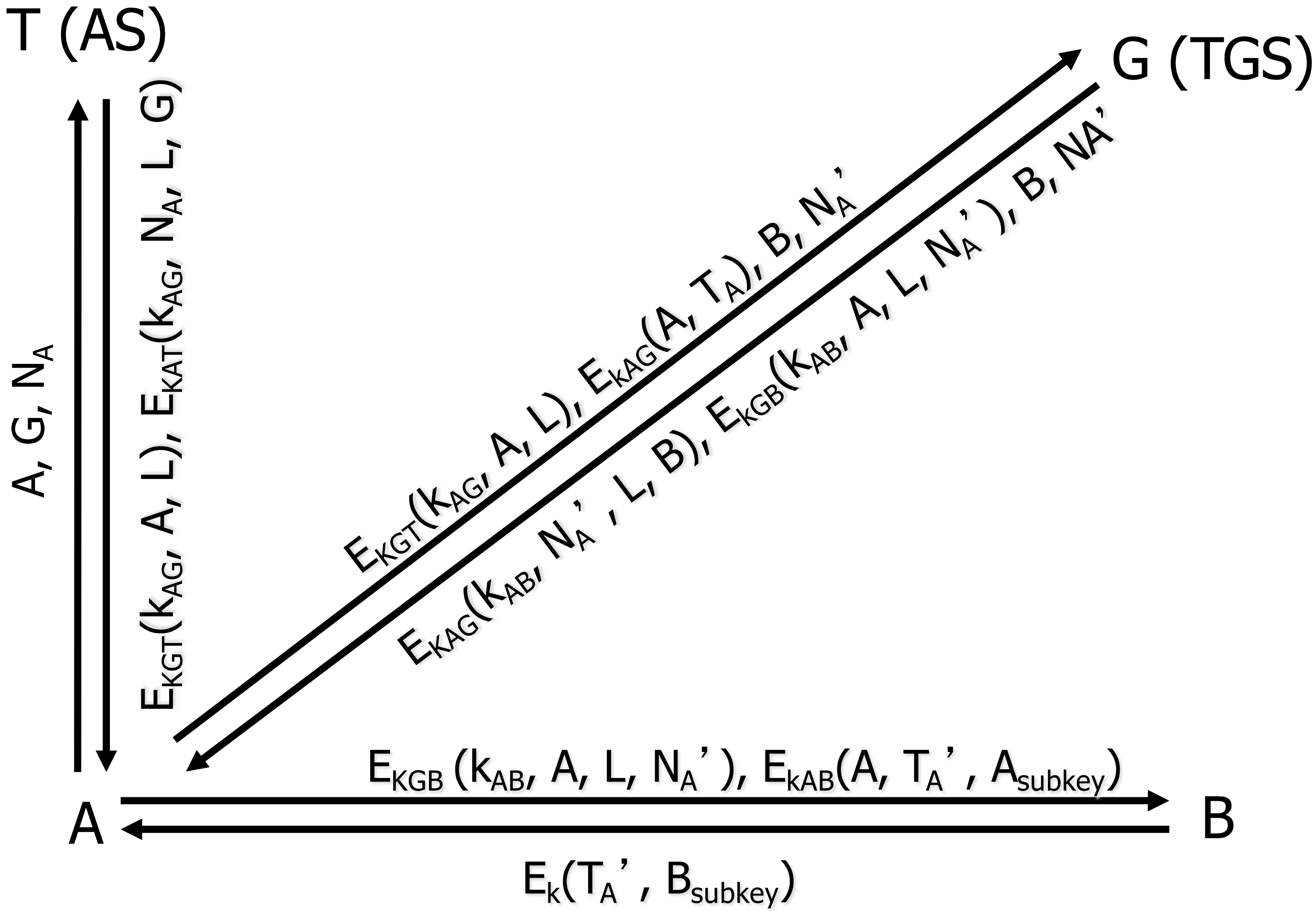
- Still debating ☺
- Let’s see one by one!

Kerberos (cnt.)

- $E_{KBT}(k, A, L)$: Token for B
- $E_{KAT}(k, N_A, L, B)$: Token for A
- L: Life-time
- N_A?
- $E_k(A, T_A, A_{subkey})$: To prove B that A knows k
- T_A: Time-stamp
- $E_k(B, T_A, B_{subkey})$: To prove A that B knows k



Kerberos (Scalable)



Public Key Certificate

- Public-key certificates are a vehicle
  - public keys may be stored, distributed or forwarded over unsecured media
- The objective
  - make one entity's public key available to others such that its authenticity and validity are verifiable.
- A public-key certificate is a data structure
  - data part
    - » cleartext data including a public key and a string identifying the party (subject entity) to be associated therewith.
  - signature part
    - » digital signature of a certification authority over the data part
    - » binding the subject entity's identity to the specified public key.

CA

- a trusted third party whose signature on the certificate vouches for the authenticity of the public key bound to the subject entity
  - The significance of this binding must be provided by additional means, such as an attribute certificate or policy statement.
- the subject entity must be a unique name within the system (distinguished name)
- The CA requires its own signature key pair, the authentic public key.
- Can be off-line!

ID-based Cryptography

- No public key
- Public key = ID (email, name, etc.)
- PKG
  - Private key generation center
  - $SK_{ID} = PKG_S(ID)$
  - PKG's public key is public.
  - distributes private key associated with the ID
- Encryption: $C= E_{ID}(M)$
- Decryption: $D_{SK}(C) = M$

Discussion (PKI vs. Kerberos vs. IBE)

- On-line vs. off-line TTP
  - Implication?
- Non-reputation?
- Revocation?
- Scalability?
- Trust issue?

Questions?

- Yongdae Kim
  - email: yongdaek@kaist.ac.kr
  - Home: http://syssec.kaist.ac.kr/~yongdaek
  - Facebook: https://www.facebook.com/y0ngdaek
  - Twitter: https://twitter.com/yongdaek
  - Google "Yongdae Kim"