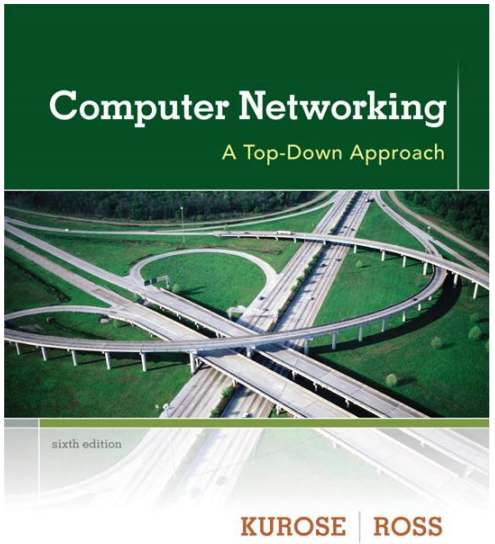
Network Security



Computer Networking: A Top Down Approach
6th edition
Jim Kurose, Keith Ross
Addison-Wesley
March 2012

A note on the use of these ppt slides:

The notes used in this course are substantially based on powerpoint slides developed and copyrighted by J.F. Kurose and K.W. Ross, 2012

Outline

- General topics in network security
- Wireless Security
- IoT Security

Roadmap

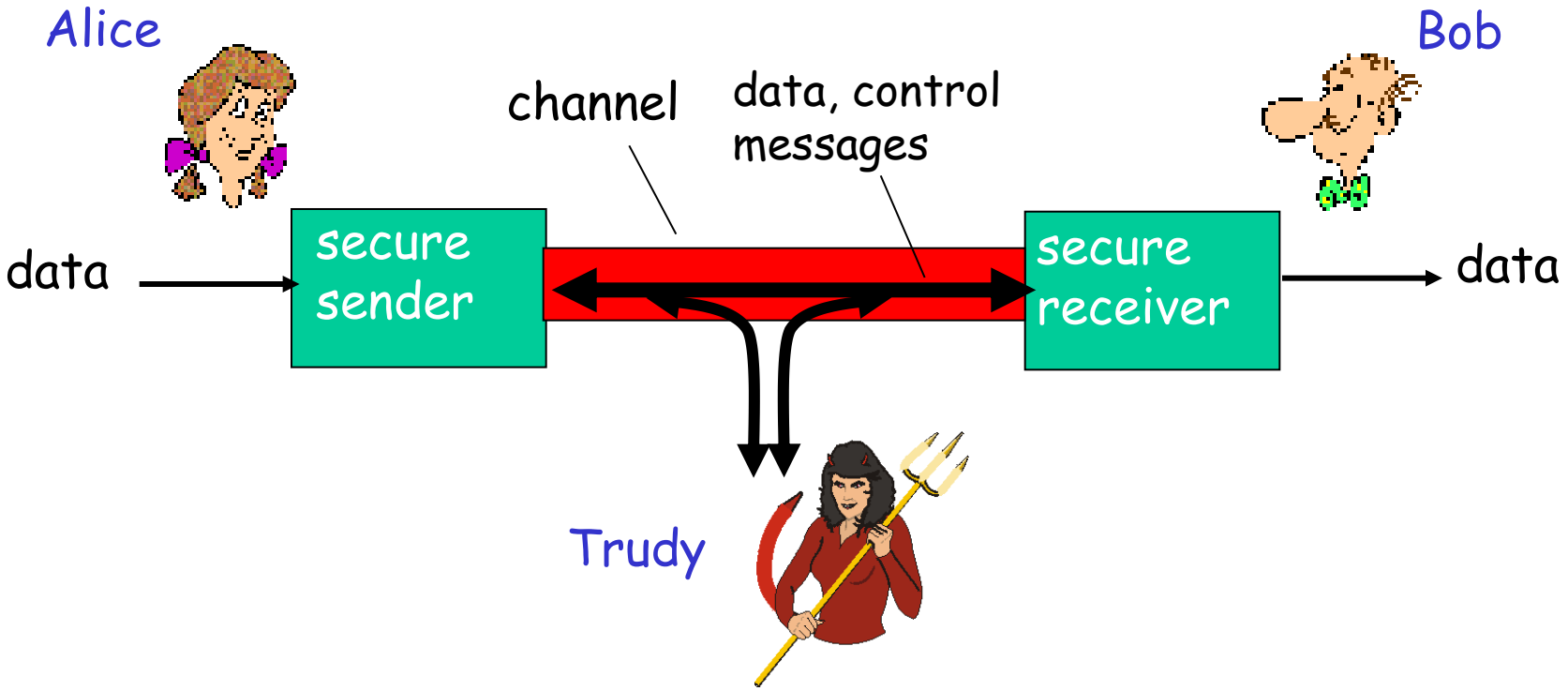
1. What is network security?
2. Principles of cryptography
3. Message integrity
4. End point authentication
5. Securing e-mail
6. Securing TCP connections: SSL
7. Network layer security: IPsec
8. Operational security: firewalls and IDS

What is network security?

- Confidentiality: only sender, intended receiver should "understand" message contents
  - sender encrypts message
  - receiver decrypts message
- Authentication: sender, receiver want to confirm identity of each other
- Message integrity: sender, receiver want to ensure message not altered (in transit, or afterwards) without detection
- Access and availability: services must be accessible and available to users

Friends and enemies: Alice, Bob, Trudy

- well-known in network security world
- Bob, Alice (lovers!) want to communicate "securely"
- Trudy (intruder) may intercept, delete, add messages



Who might Bob, Alice be?

- … well, *real-life* Bobs and Alices!
- Web browser/server for electronic transactions (e.g., on-line purchases)
- on-line banking client/server
- DNS servers
- routers exchanging routing table updates
- other examples?

There are bad guys (and girls) out there!

Q: What can a "bad guy" do?

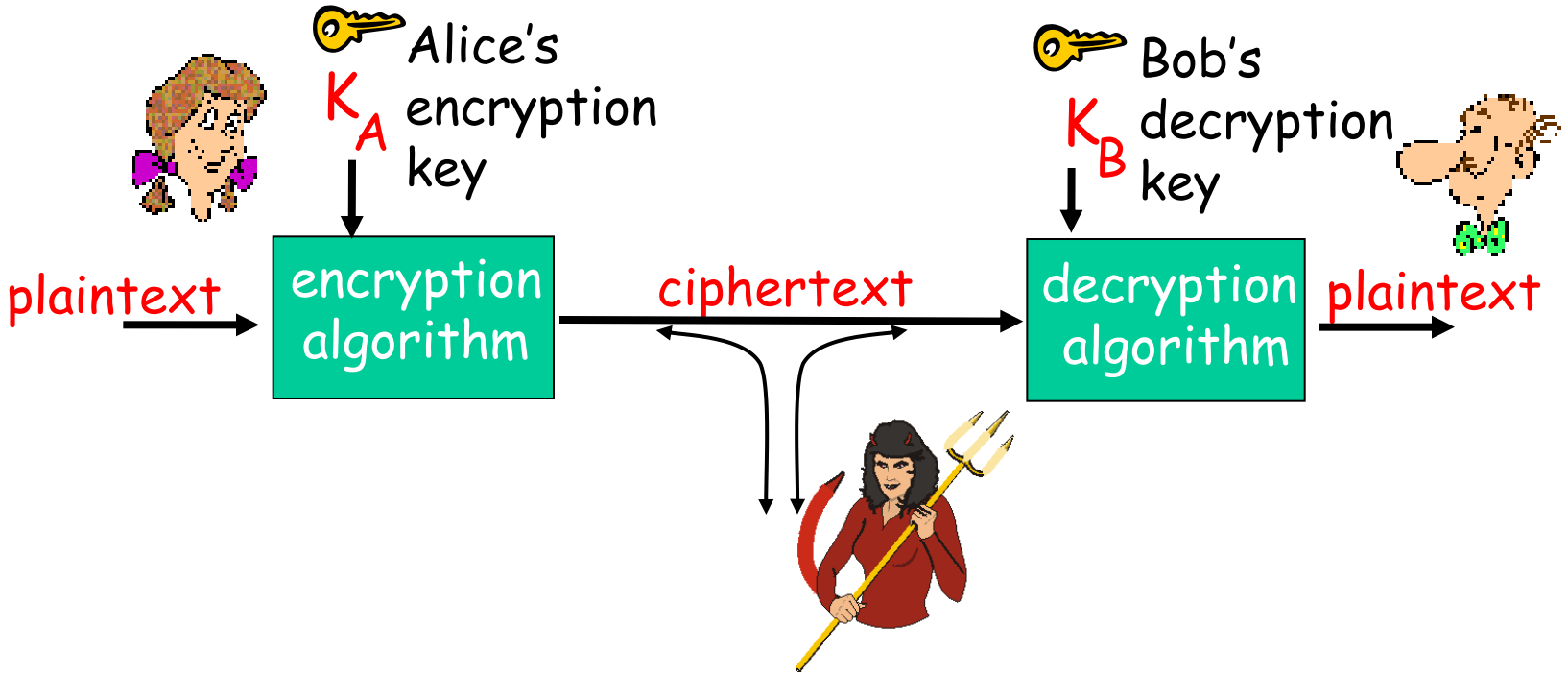
A: A lot!

- *eavesdrop:* intercept messages
- actively *insert* messages into connection
- *impersonation:* can fake (spoof) source address in packet (or any field in packet)
- *hijacking:* "take over" ongoing connection by removing sender or receiver, inserting himself in place
- *denial of service*: prevent service from being used by others (e.g., by overloading resources)

Roadmap

1. What is network security?
2. Principles of cryptography
3. Message integrity
4. End point authentication
5. Securing e-mail
6. Securing TCP connections: SSL
7. Network layer security: IPsec
8. Operational security: firewalls and IDS

The language of cryptography



m plaintext message

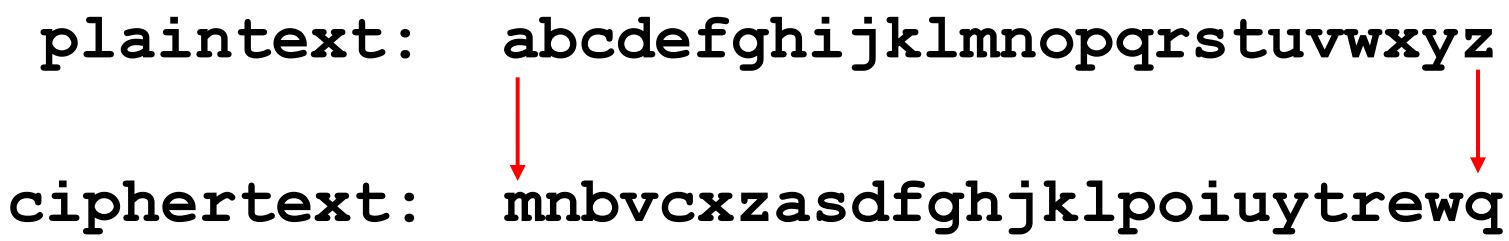
$K_A(m)$ ciphertext, encrypted with key K_A

$m = K_B(K_A(m))$

Simple encryption scheme

substitution cipher: substituting one thing for another

- monoalphabetic cipher: substitute one letter for another



E.g.: Plaintext: bob. i love you. alice

ciphertext: nkn. s gktc wky. mgsbc

Key: the mapping from the set of 26 letters to the set of 26 letters

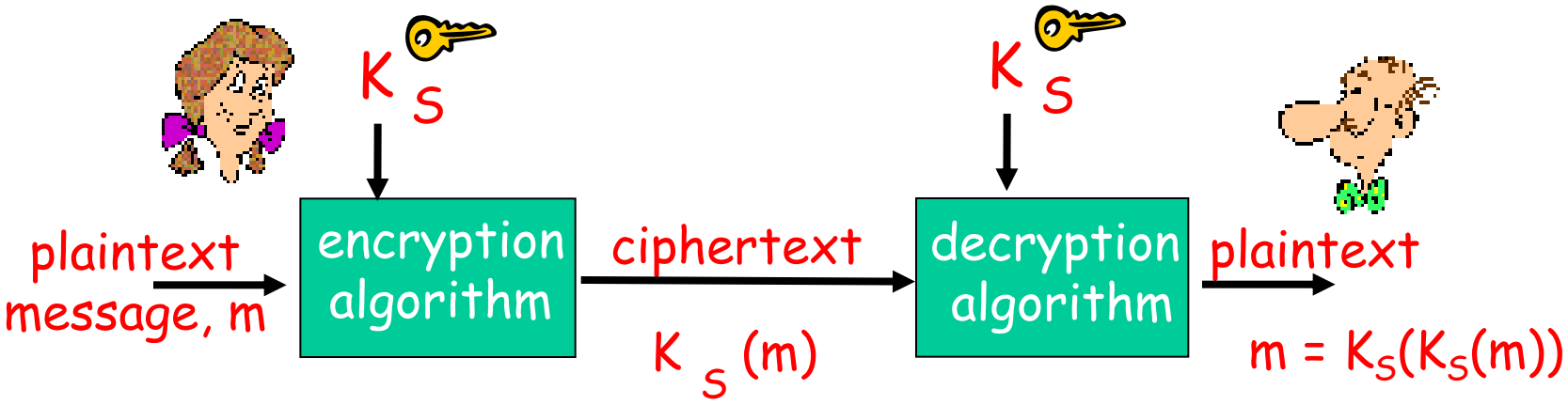
Polyalphabetic encryption

- n monoalphabetic cyphers, $M_1, M_2, \ldots, M_n$
- Cycling pattern:
  - e.g., n=4, M_1, M_3, M_4, M_3, M_2; M_1, M_3, M_4, M_3, M_2;
- For each new plaintext symbol, use subsequent monoalphabetic pattern in cyclic pattern
  - dog: d from M_1, o from M_3, g from M_4
- Key: the n ciphers and the cyclic pattern

Types of Cryptography

- Crypto often uses keys:
  - Algorithm is known to everyone
  - Only "keys" are secret
- Public key cryptography
  - Involves the use of two keys
- Symmetric key cryptography
  - Involves the use one key
- Hash functions
  - Involves the use of no keys
  - Nothing secret: How can this be useful?

Symmetric key cryptography



symmetric key crypto: Bob and Alice share same (symmetric) key: K_S

- e.g., key is knowing substitution pattern in mono alphabetic substitution cipher

Q: how do Bob and Alice agree on key value?

Two types of symmetric ciphers

- Stream ciphers
  - encrypt one bit at time
- Block ciphers
  - Break plaintext message in equal-size blocks
  - Encrypt each block as a unit

Block ciphers

- Message to be encrypted is processed in blocks of k bits (e.g., 64-bit blocks).
- 1-to-1 mapping is used to map k-bit block of plaintext to k-bit block of ciphertext

Example with k=3:

input	output
000	110
001	111
010	101
011	100

input	output
100	011
101	010
110	000
111	001

What is the ciphertext for 010110001111 ?

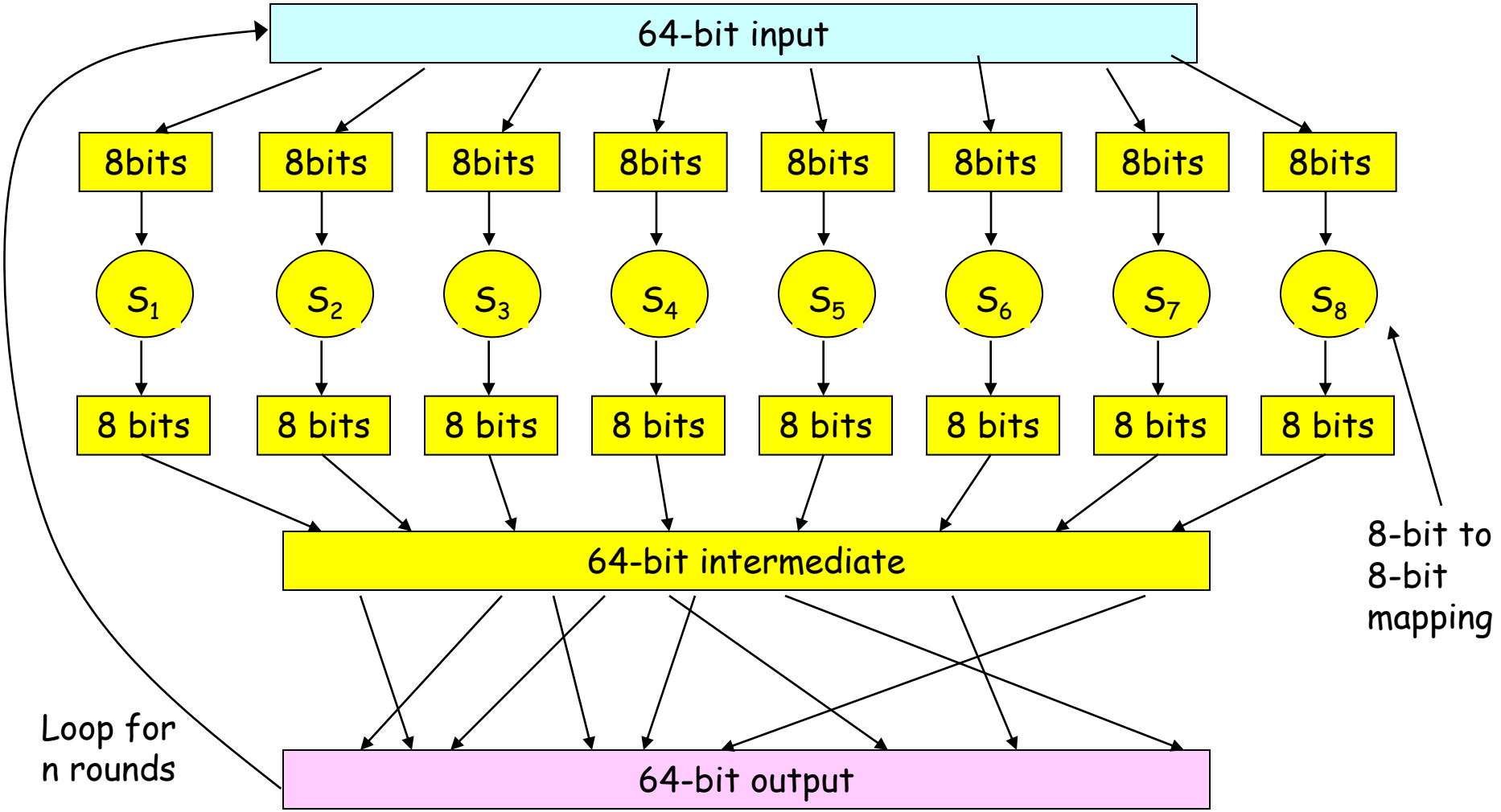
101000111001

Block ciphers

- How many possible mappings are there for k=3?
  - How many 3-bit inputs?
  - How many permutations of the 3-bit inputs?
  - Answer: 40,320 ; not very many!
- In general, $2^k!$ mappings; huge for k=64
- Problem:
  - Table approach requires table with 2^{64} entries, each entry with 64 bits
- Table too big: instead use function that simulates a randomly permuted table

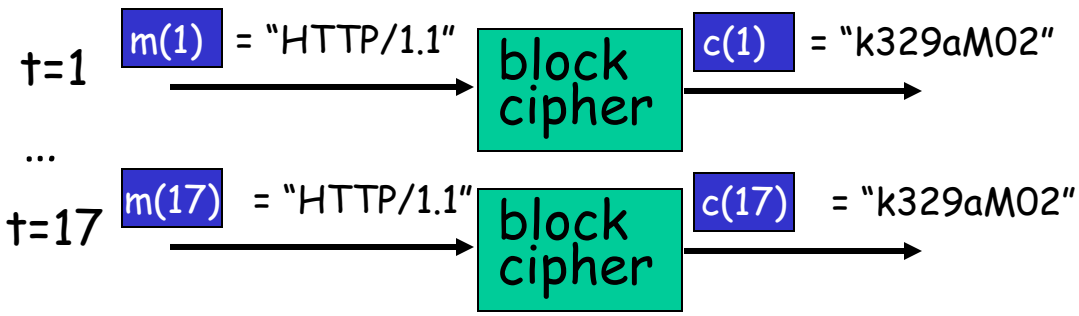
Prototype function

From Kaufman et al

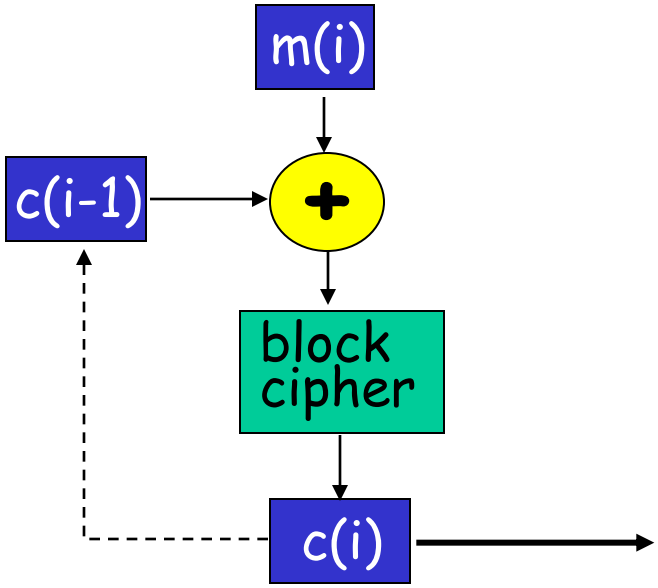


Cipher Block Chaining (CBC)

- cipher block: if input block repeated, will produce same cipher text:



- *cipher block chaining:* XOR ith input block, m(i), with previous block of cipher text, c(i-1)
  - c(0) transmitted to receiver in clear
  - what happens in "HTTP/1.1" scenario from above?



Symmetric key crypto: DES

DES: Data Encryption Standard

- US encryption standard [NIST 1993]
- 56-bit symmetric key, 64-bit plaintext input
- Block cipher with cipher block chaining
- How secure is DES?
  - DES Challenge: 56-bit-key-encrypted phrase decrypted (brute force) in less than a day
  - No known good analytic attack
- making DES more secure:
  - 3DES: encrypt 3 times with 3 different keys

(actually encrypt, decrypt, encrypt)

AES: Advanced Encryption Standard

- new (Nov. 2001) symmetric-key NIST standard, replacing DES
- processes data in 128 bit blocks
- 128, 192, or 256 bit keys
- brute force decryption (try each key) taking 1 sec on DES, takes 149 trillion years for AES

Public Key Cryptography

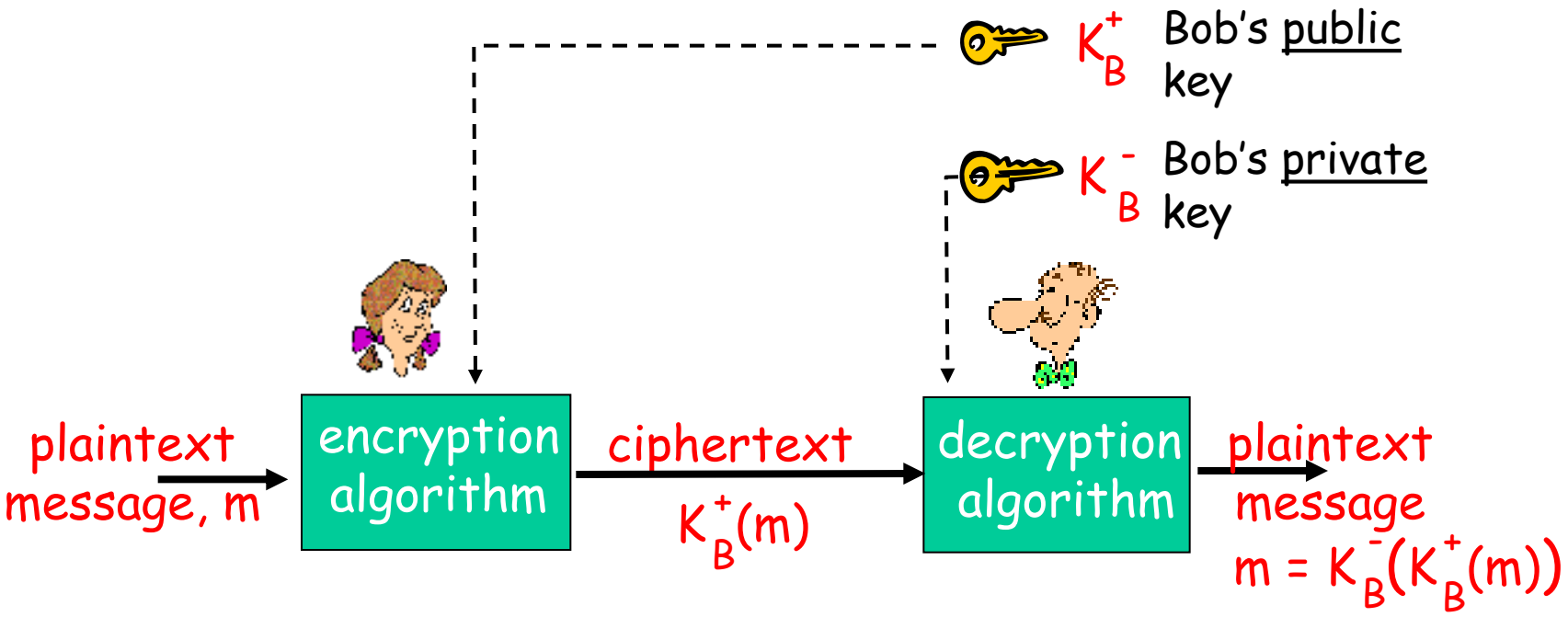
symmetric key crypto

- requires sender, receiver know shared secret key
- Q: how to agree on key in first place (particularly if never "met")?

public key cryptography

- radically different approach [Diffie-Hellman76, RSA78]
- sender, receiver do *not* share secret key
- *public* encryption key known to *all*
- *private* decryption key known only to receiver

Public key cryptography



Public key encryption algorithms

Requirements:

1. need $K_B^+(\cdot)$ and $K_B^-(\cdot)$ such that

$$K_B^-(K_B^+(m)) = m$$

2. given public key K_B^+, it should be impossible to compute private key K_B^-

RSA: Rivest, Shamir, Adelson algorithm

Prerequisite: modular arithmetic

- x mod n = remainder of x when divide by n
- Facts:

  [(a mod n) + (b mod n)] mod n = (a+b) mod n
  [(a mod n) - (b mod n)] mod n = (a-b) mod n
  [(a mod n) * (b mod n)] mod n = (a*b) mod n
- Thus

  $(a \bmod n)^d \bmod n = a^d \bmod n$
- Example: x=14, n=10, d=2:
  $(x \bmod n)^d \bmod n = 14^2 \bmod 10 = 6$
  $x^d = 14^2 = 196 \quad x^d \bmod 10 = 6$

RSA: getting ready

- A message is a bit pattern.
- A bit pattern can be uniquely represented by an integer number.
- Thus encrypting a message is equivalent to encrypting a number.

Example

- m= 10010001 . This message is uniquely represented by the decimal number 145.
- To encrypt m, we encrypt the corresponding number, which gives a new number (the cyphertext).

RSA: Creating public/private key pair

1. Choose two large prime numbers *p*, *q*. (e.g., 1024 bits each)

2. Compute *n* = *pq*, *z* = *(p-1)(q-1)*

3. Choose *e* (with *e<n*) that has no common factors with z. (*e*, *z* are "relatively prime").

4. Choose *d* such that *ed-1* is exactly divisible by *z*. (in other words: *ed* mod *z* = *1*).

5. *Public* key is *(n,e)*. *Private* key is *(n,d)*.

K_B^+ K_B^-

RSA: Encryption, decryption

0. Given (n,e) and (n,d) as computed above

1. To encrypt message m ($<n$), compute

   $c = m^e \bmod n$

2. To decrypt received bit pattern, c, compute

   $m = c^d \bmod n$

Magic happens!

$$m = \underbrace{(m^e \bmod n)}_{c}{}^d \bmod n$$

RSA example:

Bob chooses *p=5, q=7*. Then *n=35, z=24*.

e=5 (so *e, z* relatively prime).

d=29 (so *ed-1* exactly divisible by z).

Encrypting 8-bit messages.

	bit pattern	m	m^e	$c = m^e \bmod n$
encrypt:	0000l000	12	24832	17

	c	c^d	$m = c^d \bmod n$
decrypt:	17	481968572106750915091411825223071697	12

Why does RSA work?

- Must show that $c^d \bmod n = m$ where $c = m^e \bmod n$
- Fact: for any x and y: $x^y \bmod n = x^{(y \bmod z)} \bmod n$
  - where n= pq and z = (p-1)(q-1)
- Thus,

$$
\begin{aligned}
c^d \bmod n &= (m^e \bmod n)^d \bmod n \\
&= m^{ed} \bmod n \\
&= m^{(ed \bmod z)} \bmod n \\
&= m^1 \bmod n \\
&= m
\end{aligned}
$$

RSA: another important property

The following property will be *very* useful later:

$$\underbrace{K_B^-(K_B^+(m))}_{\text{use public key first, followed by private key}} = m = \underbrace{K_B^+(K_B^-(m))}_{\text{use private key first, followed by public key}}$$

Result is the same!

Why $K_B^-(K_B^+(m)) = m = K_B^+(K_B^-(m))$?

Follows directly from modular arithmetic:

$$(m^e \bmod n)^d \bmod n = m^{ed} \bmod n$$
$$= m^{de} \bmod n$$
$$= (m^d \bmod n)^e \bmod n$$

Why is RSA Secure?

- Suppose you know Bob's public key (n,e). How hard is it to determine d?
- Essentially need to find factors of n without knowing the two factors p and q.
- Fact: factoring a big number is hard.

Generating RSA keys

- Have to find big primes p and q
- Approach: make good guess then apply testing rules (see Kaufman)

Session keys

- Exponentiation is computationally intensive
- DES is at least 100 times faster than RSA

Session key, K_S

- Bob and Alice use RSA to exchange a symmetric key K_S
- Once both have K_S, they use symmetric key cryptography

Roadmap

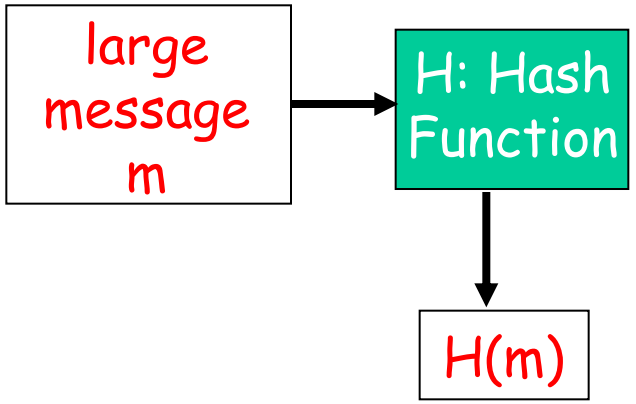
1. What is network security?
2. Principles of cryptography
3. Message integrity
4. End point authentication
5. Securing e-mail
6. Securing TCP connections: SSL
7. Network layer security: IPsec
8. Operational security: firewalls and IDS

Message Integrity

- Allows communicating parties to verify that received messages are authentic.
  - Content of message has not been altered
  - Source of message is who/what you think it is
  - Message has not been replayed
  - Sequence of messages is maintained
- Let's first talk about message digests

Message Digests

- Function H( ) that takes as input an arbitrary length message and outputs a fixed-length string: "message signature"
- Note that H( ) is a many-to-1 function
- H( ) is often called a "hash function"



- Desirable properties:
  - Easy to calculate
  - Irreversibility: Can't determine m from H(m)
  - Collision resistance: Computationally difficult to produce m and m' such that H(m) = H(m')
  - Seemingly random output

Internet checksum: poor message digest

Internet checksum has some properties of hash function:

- ✓ produces fixed length digest (16-bit sum) of input
- ✓ is many-to-one

- But given message with given hash value, it is easy to find another message with same hash value.
- Example: Simplified checksum: add 4-byte chunks at a time:

message	ASCII format
I O U 1	49 4F 55 31
0 0 . 9	30 30 2E 39
9 B O B	39 42 D2 42
	B2 C1 D2 AC

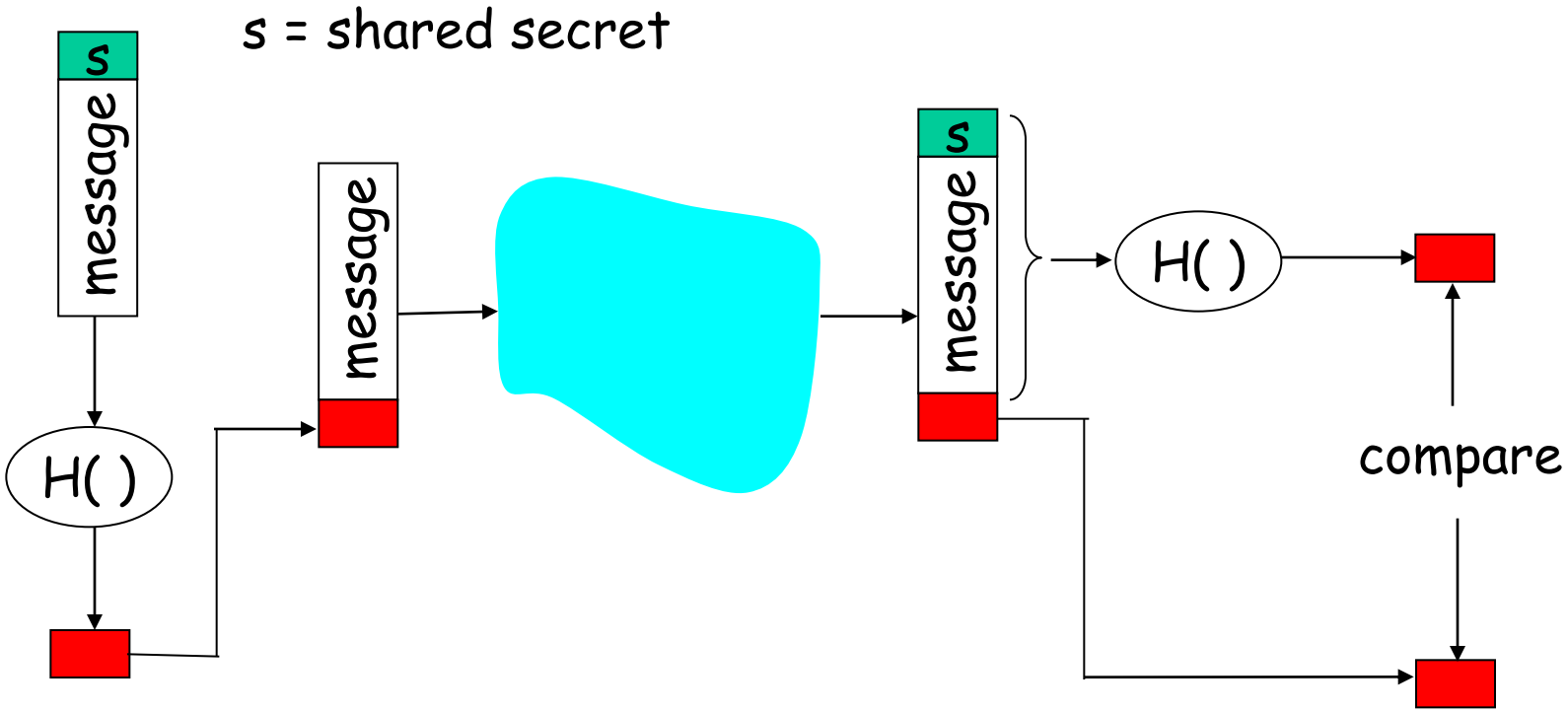
message	ASCII format
I O U 9	49 4F 55 39
0 0 . 1	30 30 2E 31
9 B O B	39 42 D2 42
	B2 C1 D2 AC

different messages but identical checksums!

Hash Function Algorithms

- MD5 hash function widely used (RFC 1321)
  - computes 128-bit message digest in 4-step process.
- SHA-1 is also used.
  - US standard [NIST, FIPS PUB 180-1]
  - 160-bit message digest

Message Authentication Code (MAC)



- ***Authenticates sender***
- ***Verifies message integrity***
- No encryption !
- Also called "keyed hash"
- Notation: $MD_m = H(s||m)$; send $m||MD_m$

Digital Signatures

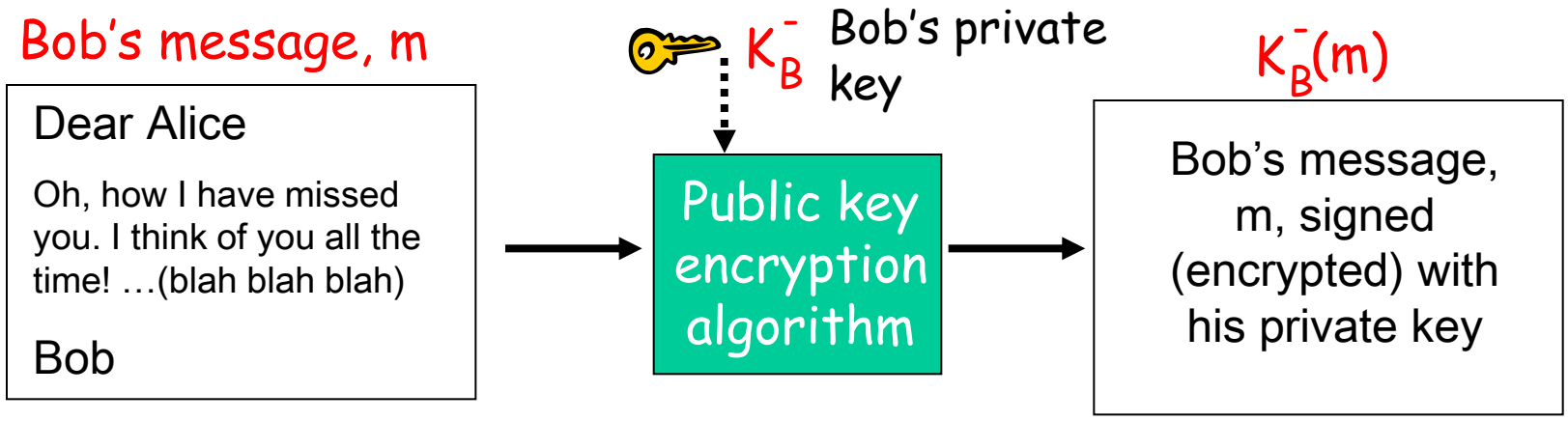
Cryptographic technique analogous to hand-written signatures.

- sender (Bob) digitally signs document, establishing he is document owner/creator.
- Goal is similar to that of a MAC, except now use public-key cryptography
- verifiable, nonforgeable: recipient (Alice) can prove to someone that Bob, and no one else (including Alice), must have signed document

Digital Signatures

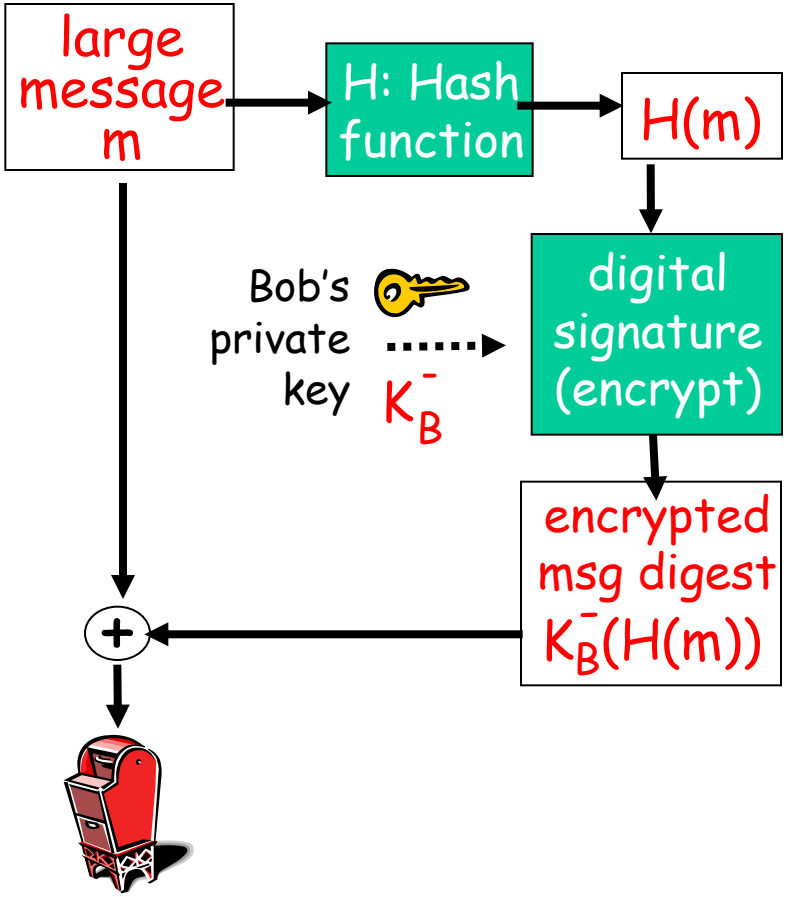
Simple digital signature for message m:

- Bob signs m by encrypting with his private key K_B^-, creating "signed" message, $K_B^-(m)$

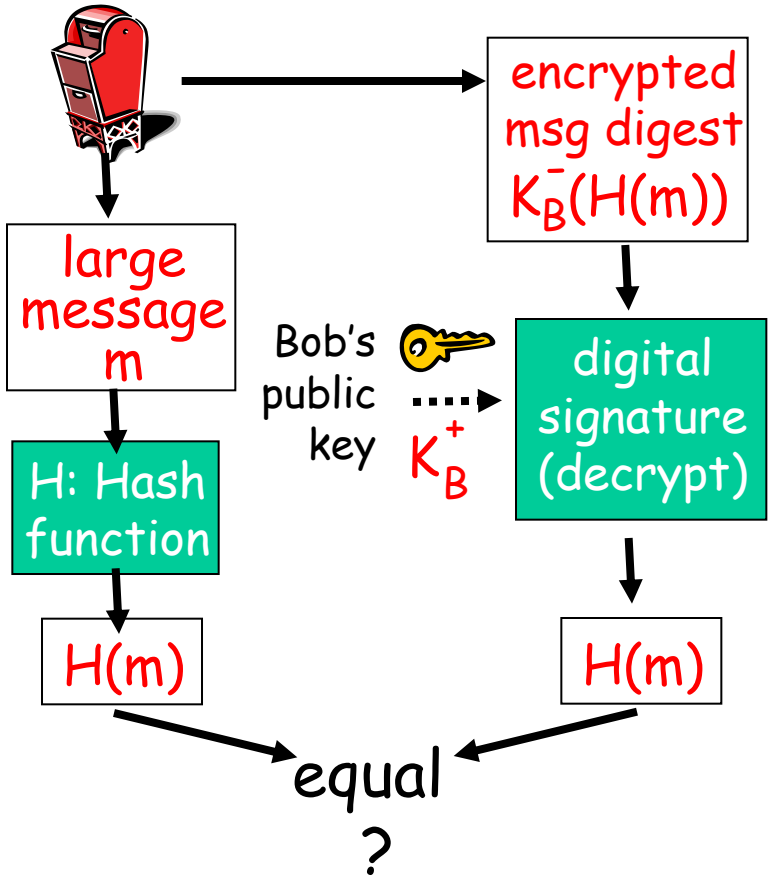


Digital signature = signed message digest

Bob sends digitally signed message:



Alice verifies signature and integrity of digitally signed message:



Digital Signatures (more)

- Suppose Alice receives msg m, digital signature $K_B^-(m)$
- Alice verifies m signed by Bob by applying Bob's public key K_B^+ to $K_B^-(m)$ then checks $K_B^+(K_B^-(m)) = m$.
- If $K_B^+(K_B^-(m)) = m$, whoever signed m must have used Bob's private key.

Alice thus verifies that:

- ✓ Bob signed m.
- ✓ No one else signed m.
- ✓ Bob signed m and not m'.

Non-repudiation:

- ✓ Alice can take m, and signature $K_B^-(m)$ to court and prove that Bob signed m.

Roadmap

1. What is network security?
2. Principles of cryptography
3. Message integrity
4. End point authentication
5. Securing e-mail
6. Securing TCP connections: SSL
7. Network layer security: Ipsec
8. Operational security: firewalls and IDS

Authentication

Goal: Bob wants Alice to "prove" her identity to him

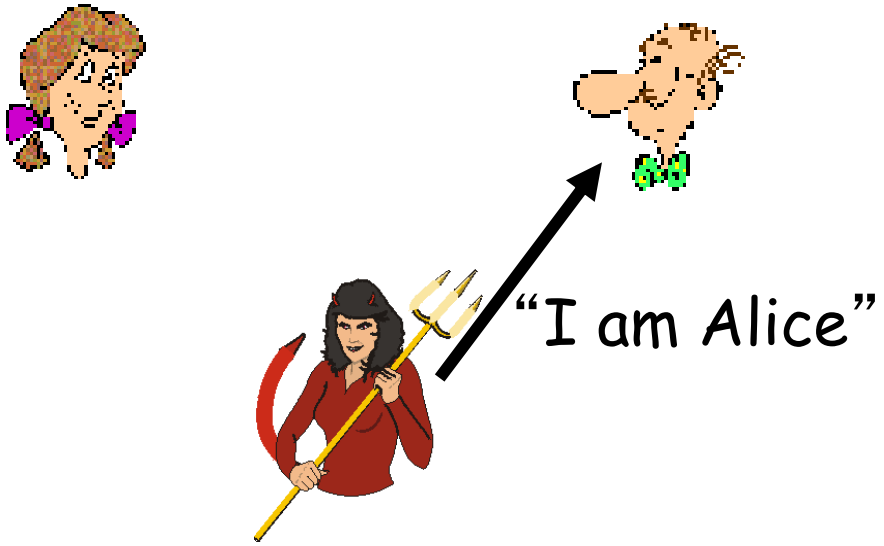
Protocol ap1.0: Alice says "I am Alice"

Failure scenario??

Authentication

Goal: Bob wants Alice to "prove" her identity to him

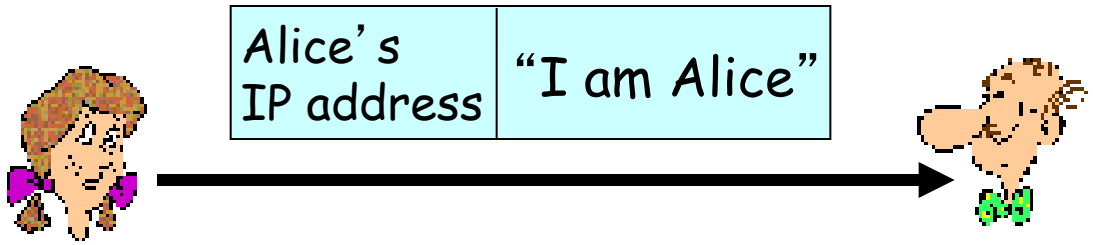
Protocol ap1.0: Alice says "I am Alice"



in a network,
Bob can not "see"
Alice, so Trudy simply
declares
herself to be Alice

Authentication: another try

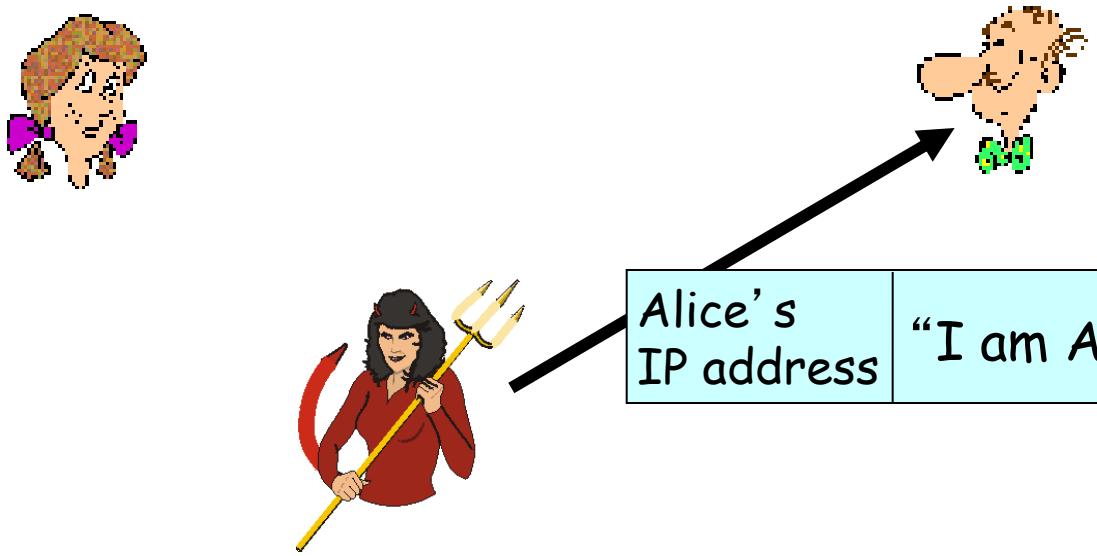
Protocol ap2.0: Alice says "I am Alice" in an IP packet containing her source IP address



Failure scenario??

Authentication: another try

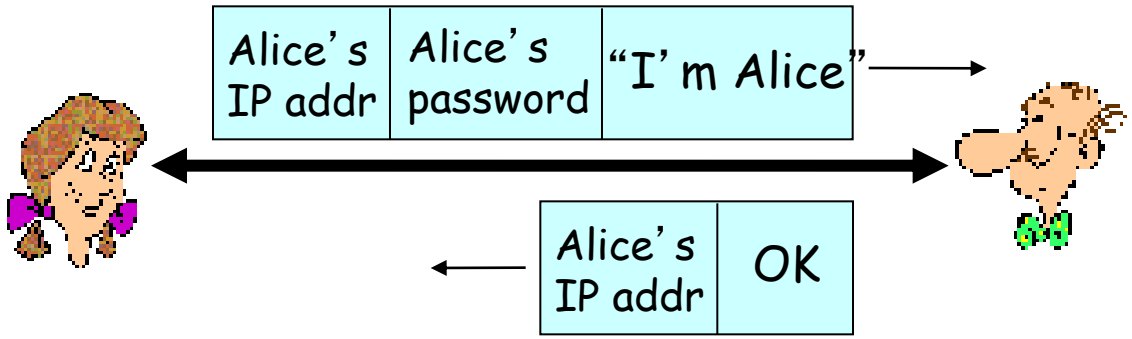
Protocol ap2.0: Alice says "I am Alice" in an IP packet containing her source IP address



Trudy can create a packet "spoofing" Alice's address

Authentication: another try

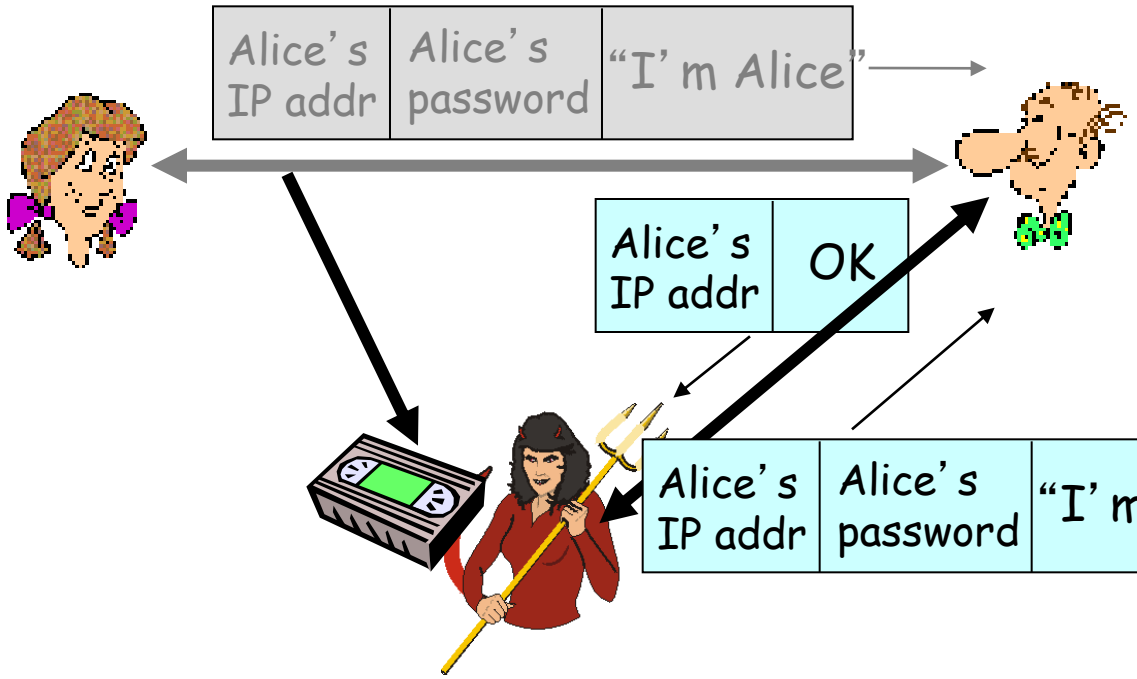
Protocol ap3.0: Alice says "I am Alice" and sends her secret password to "prove" it.



Failure scenario??

Authentication: another try

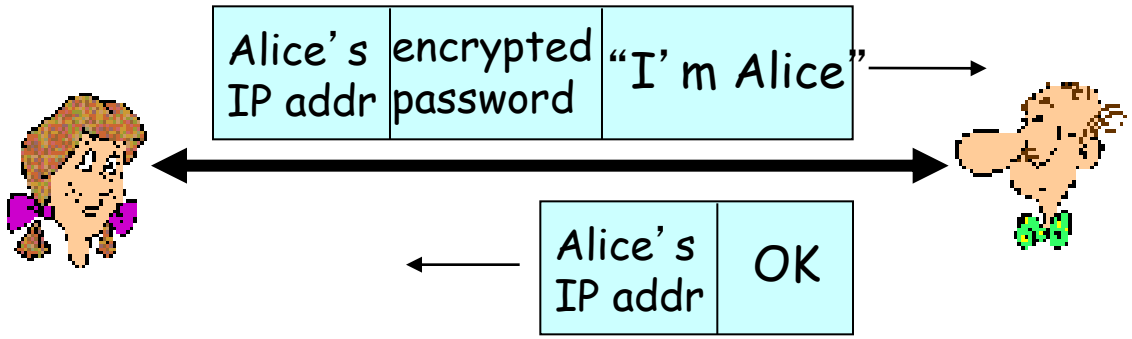
Protocol ap3.0: Alice says "I am Alice" and sends her secret password to "prove" it.



playback attack: Trudy records Alice's packet and later plays it back to Bob

Authentication: yet another try

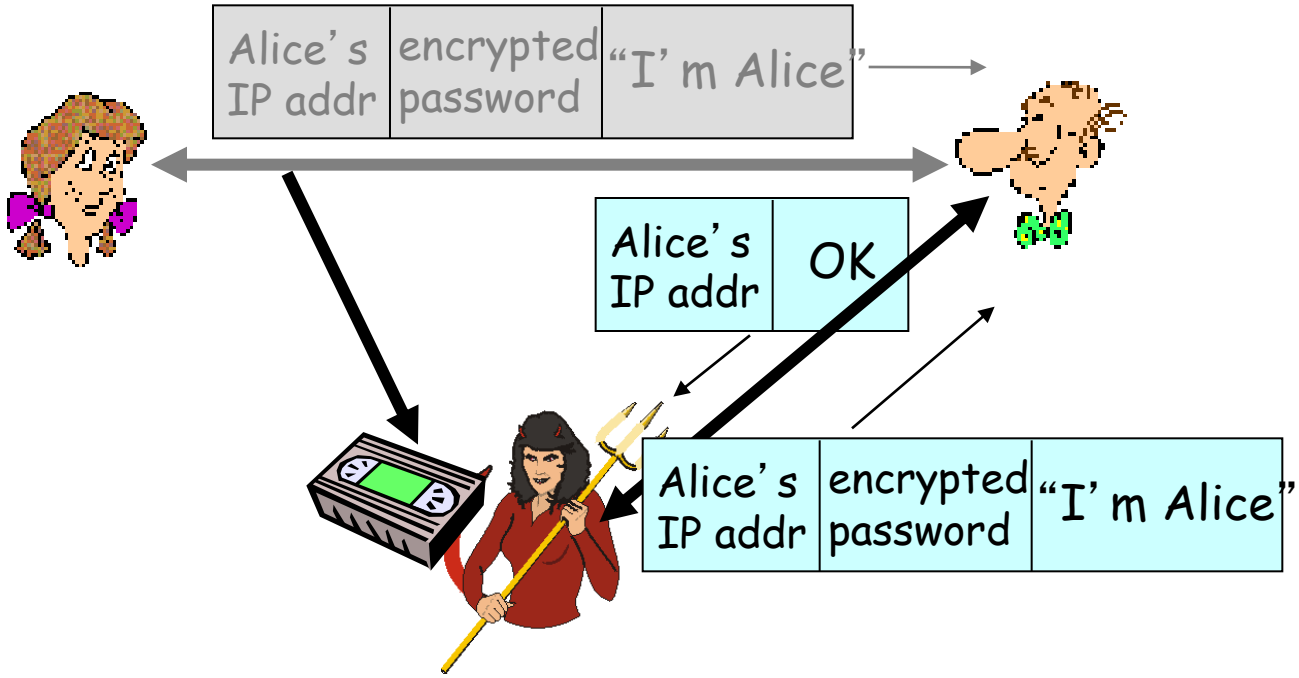
Protocol ap3.1: Alice says "I am Alice" and sends her *encrypted* secret password to "prove" it.



Failure scenario??

Authentication: another try

Protocol ap3.1: Alice says "I am Alice" and sends her *encrypted* secret password to "prove" it.



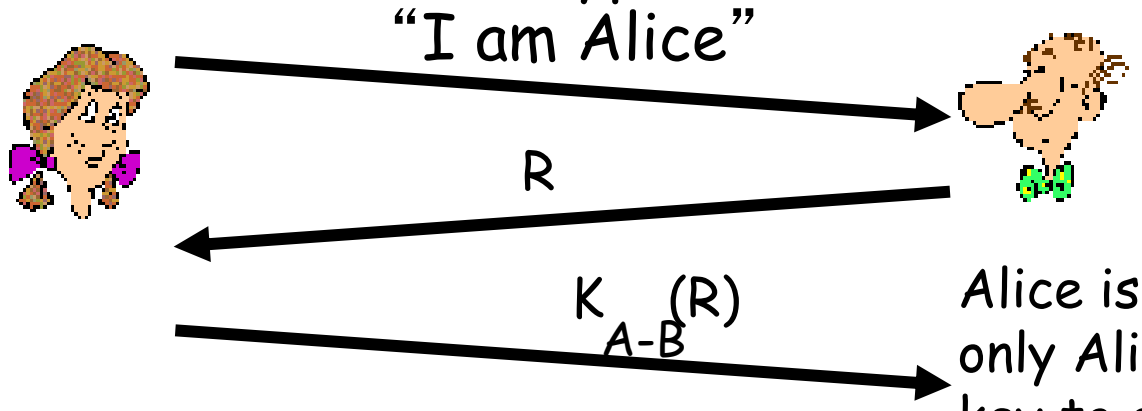
record and playback still works!

Authentication: yet another try

Goal: avoid playback attack

Nonce: number (R) used only *once –in-a-lifetime*

ap4.0: to prove Alice "live", Bob sends Alice nonce, R. Alice must return R, encrypted with shared secret key



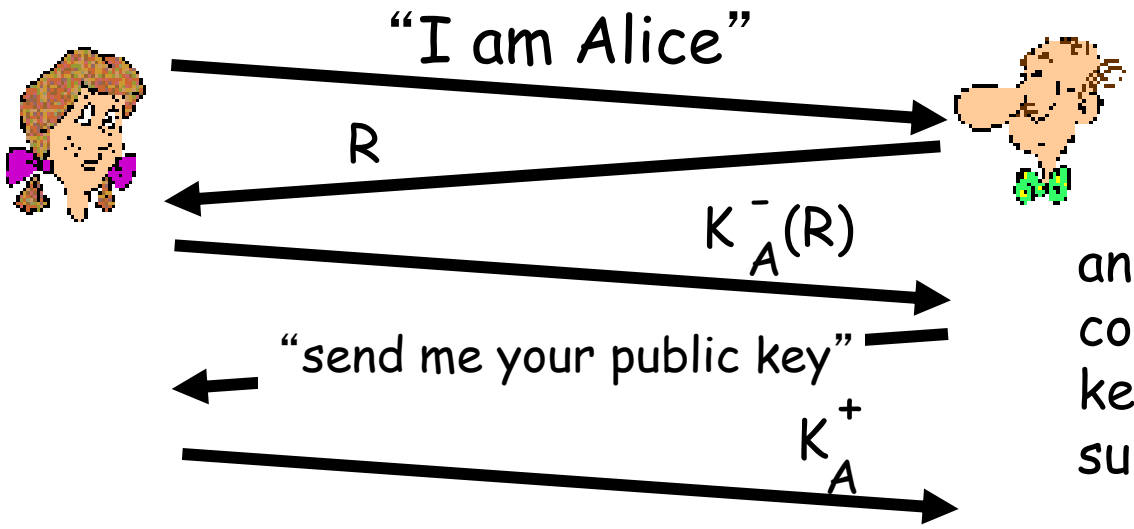
Failures, drawbacks?

Authentication: ap5.0

ap4.0 requires shared symmetric key

- can we authenticate using public key techniques?

ap5.0: use nonce, public key cryptography

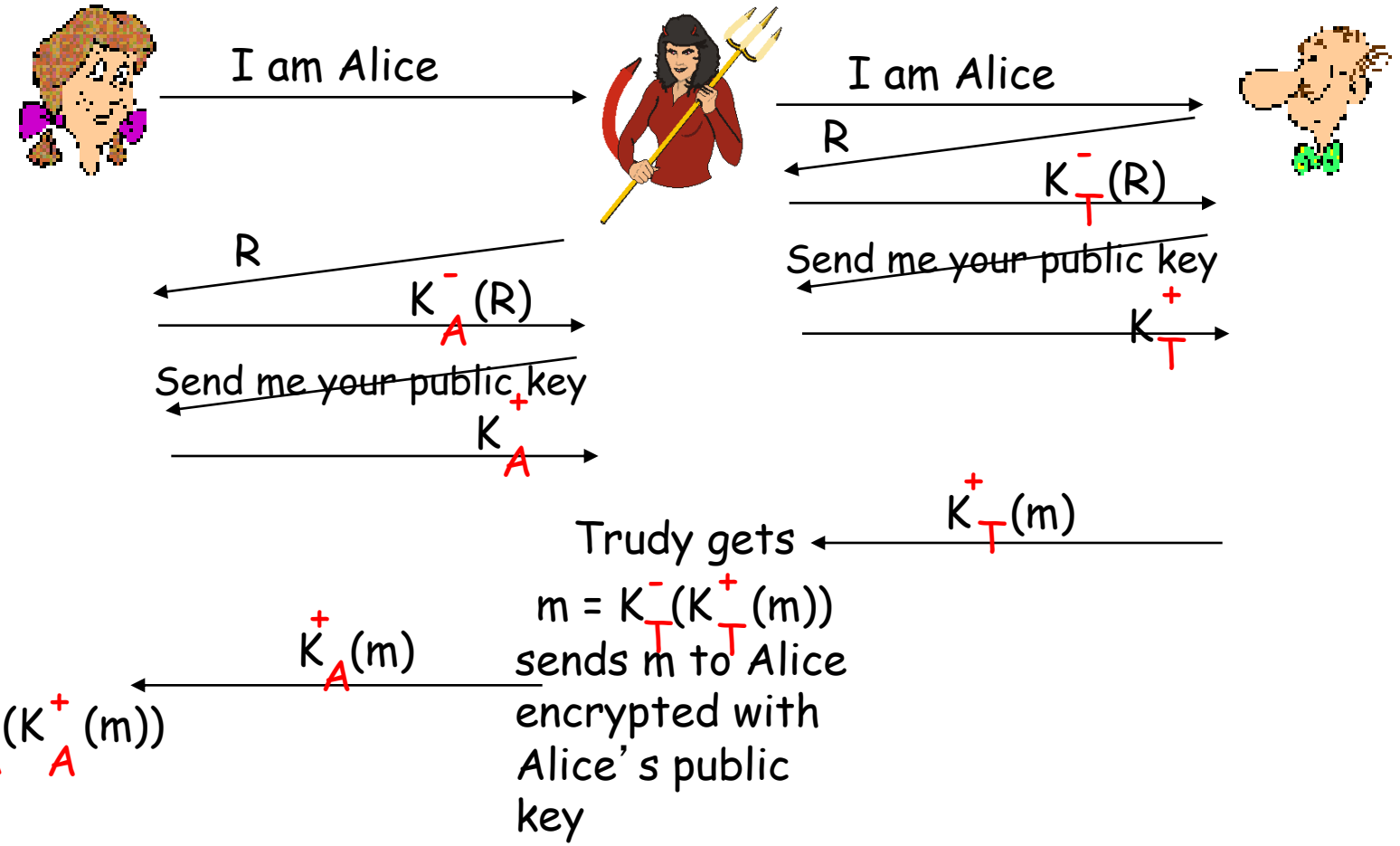


Bob computes $K_A^+(K_A^-(R)) = R$ and knows only Alice could have the private key, that encrypted R such that

$K_A^+(K_A^-(R)) = R$

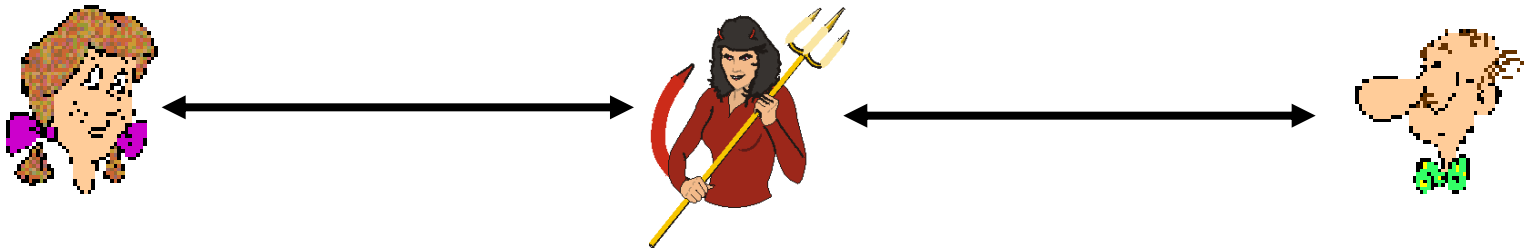
ap5.0: security hole

Man (woman) in the middle attack: Trudy poses as Alice (to Bob) and as Bob (to Alice)



ap5.0: security hole

Man (woman) in the middle attack: Trudy poses as Alice (to Bob) and as Bob (to Alice)



Difficult to detect:

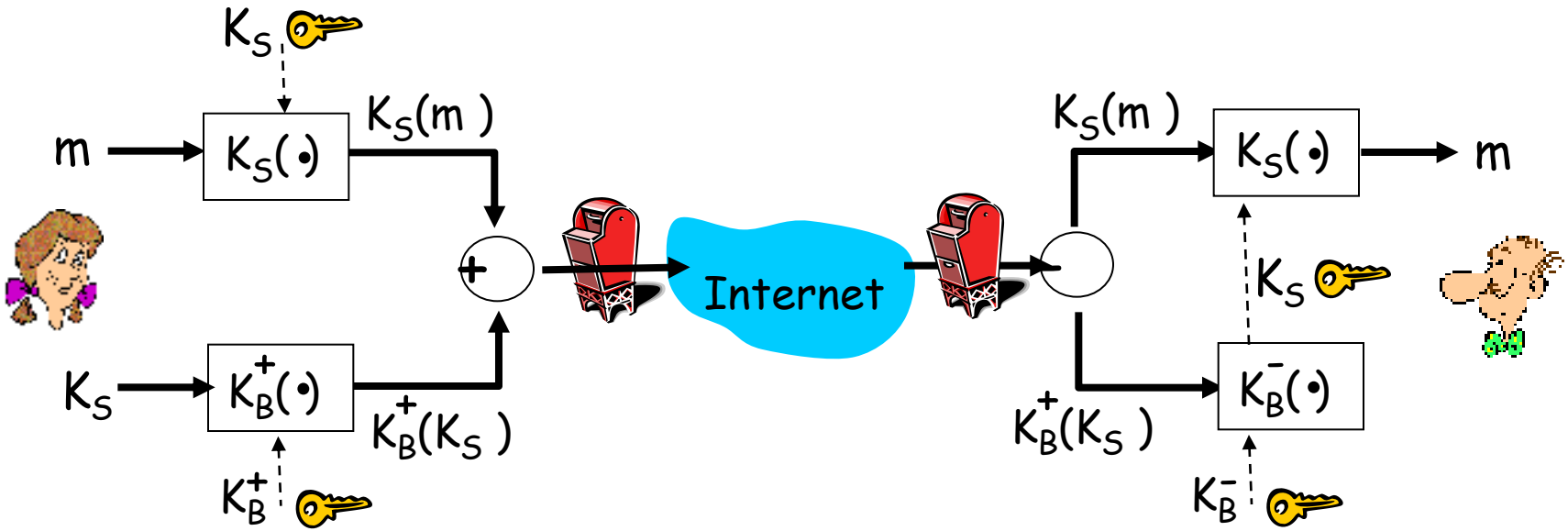
- Bob receives everything that Alice sends, and vice versa. (e.g., so Bob, Alice can meet one week later and recall conversation)
- problem is that Trudy receives all messages as well!

Roadmap

1. What is network security?
2. Principles of cryptography
3. Message integrity
4. End point authentication
5. Securing e-mail
6. Securing TCP connections: SSL
7. Network layer security: IPsec
8. Operational security: firewalls and IDS

Secure e-mail

- Alice wants to send confidential e-mail, m, to Bob.

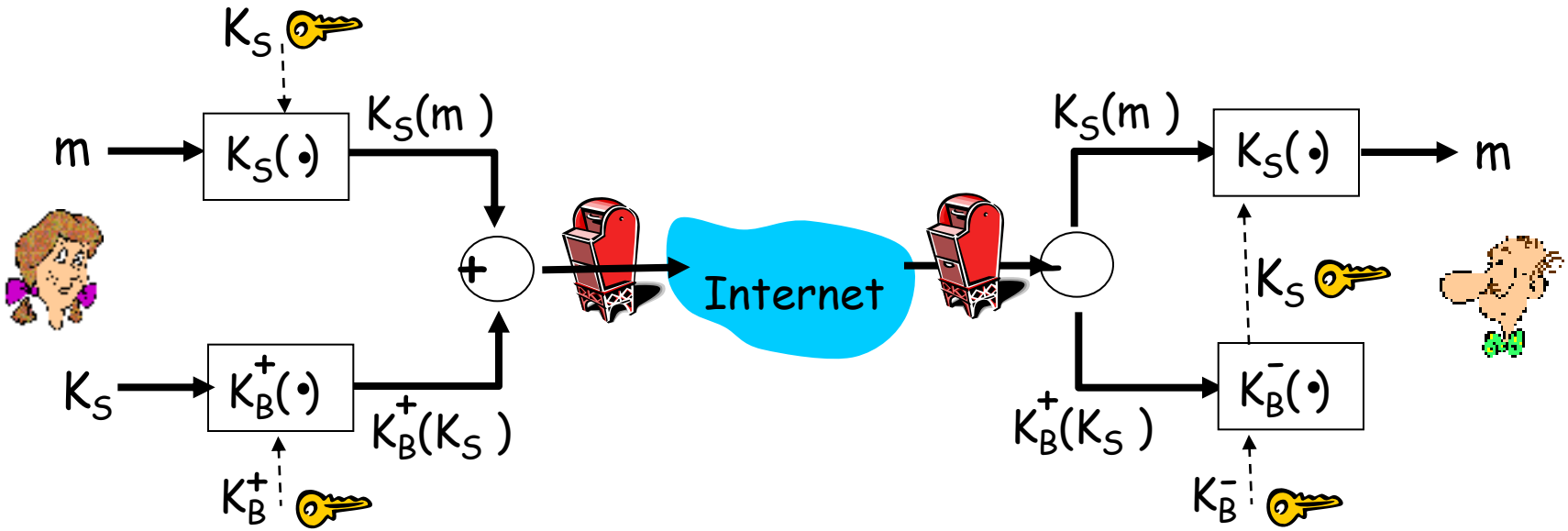


Alice:

- generates random *symmetric* private key, K_S.
- encrypts message with K_S (for efficiency)
- also encrypts K_S with Bob's public key.
- sends both $K_S(m)$ and $K_B(K_S)$ to Bob.

Secure e-mail

- Alice wants to send confidential e-mail, m, to Bob.

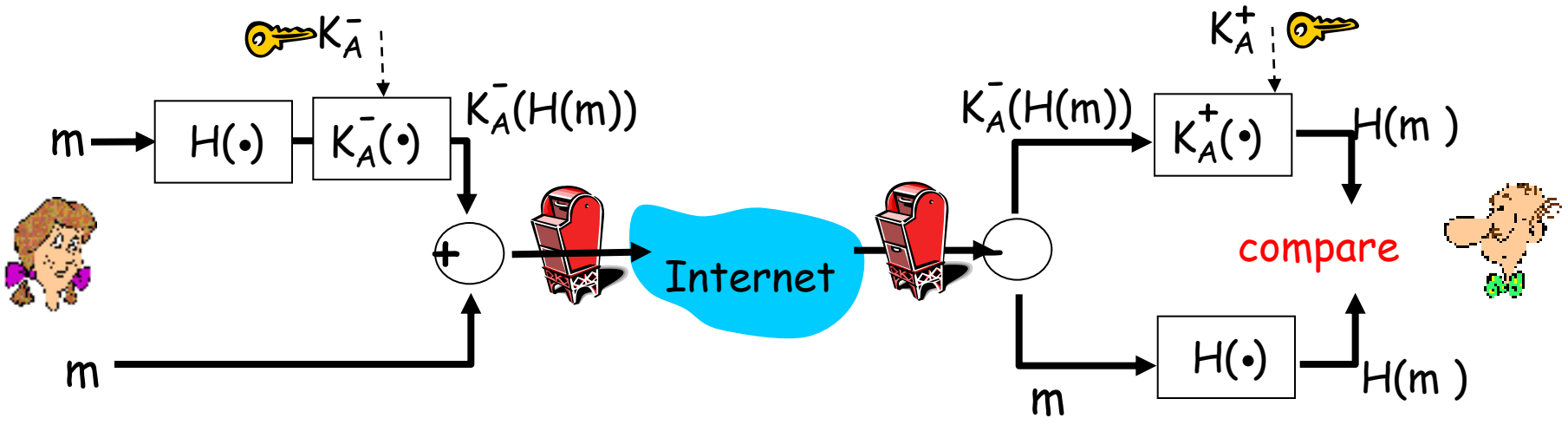


Bob:

- uses his private key to decrypt and recover K_S
- uses K_S to decrypt $K_S(m)$ to recover m

Secure e-mail (continued)

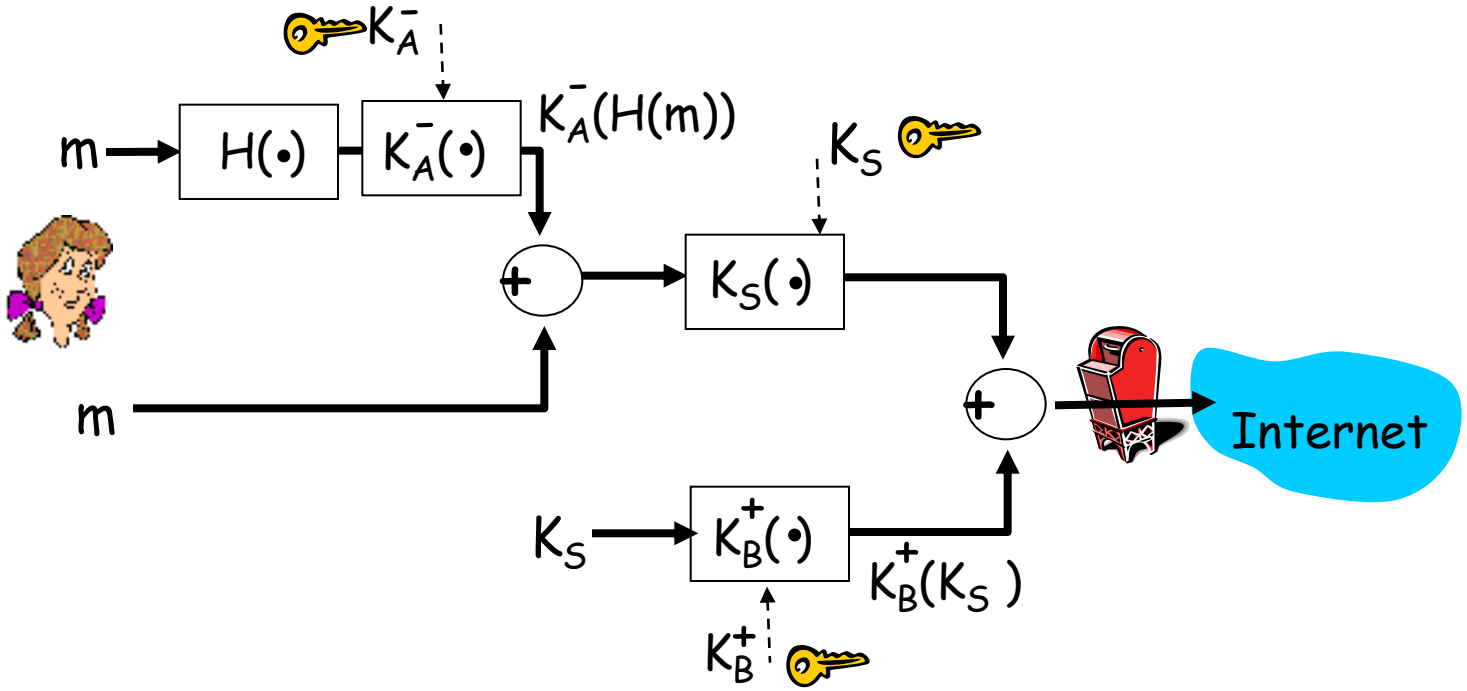
• Alice wants to provide sender authentication message integrity.



• Alice digitally signs message.
• sends both message (in the clear) and digital signature.

Secure e-mail (continued)

- Alice wants to provide secrecy, sender authentication, message integrity.



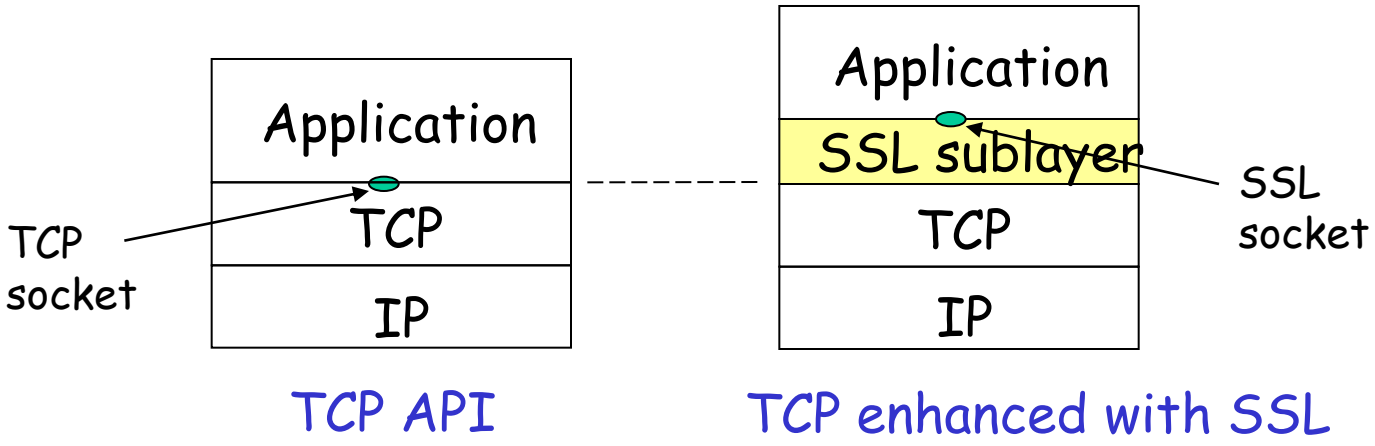
Alice uses three keys: her private key, Bob's public key, newly created symmetric key

Roadmap

1. What is network security?
2. Principles of cryptography
3. Message integrity
4. End point authentication
5. Securing e-mail
6. Securing TCP connections: SSL
7. Network layer security: IPsec
8. Operational security: firewalls and IDS

Secure sockets layer (SSL)

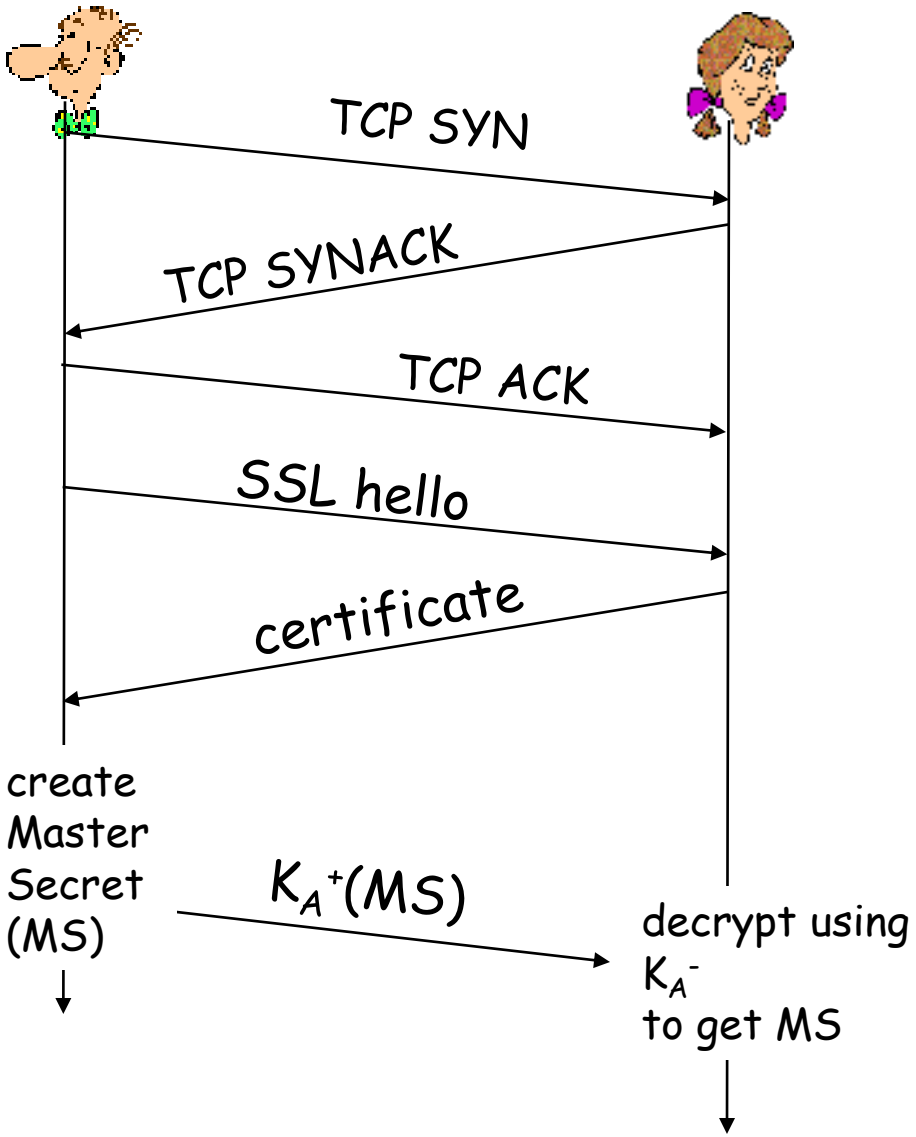
- provides transport layer security to any TCP-based application using SSL services.
  - e.g., between Web browsers, servers for e-commerce (shttp)
- security services:
  - server authentication, data encryption, client authentication (optional)



SSL: three phases

1. Handshake:

- Bob establishes TCP connection to Alice
- authenticates Alice via CA signed certificate
- creates, encrypts (using Alice's public key), sends master secret key to Alice
  - nonce exchange not shown



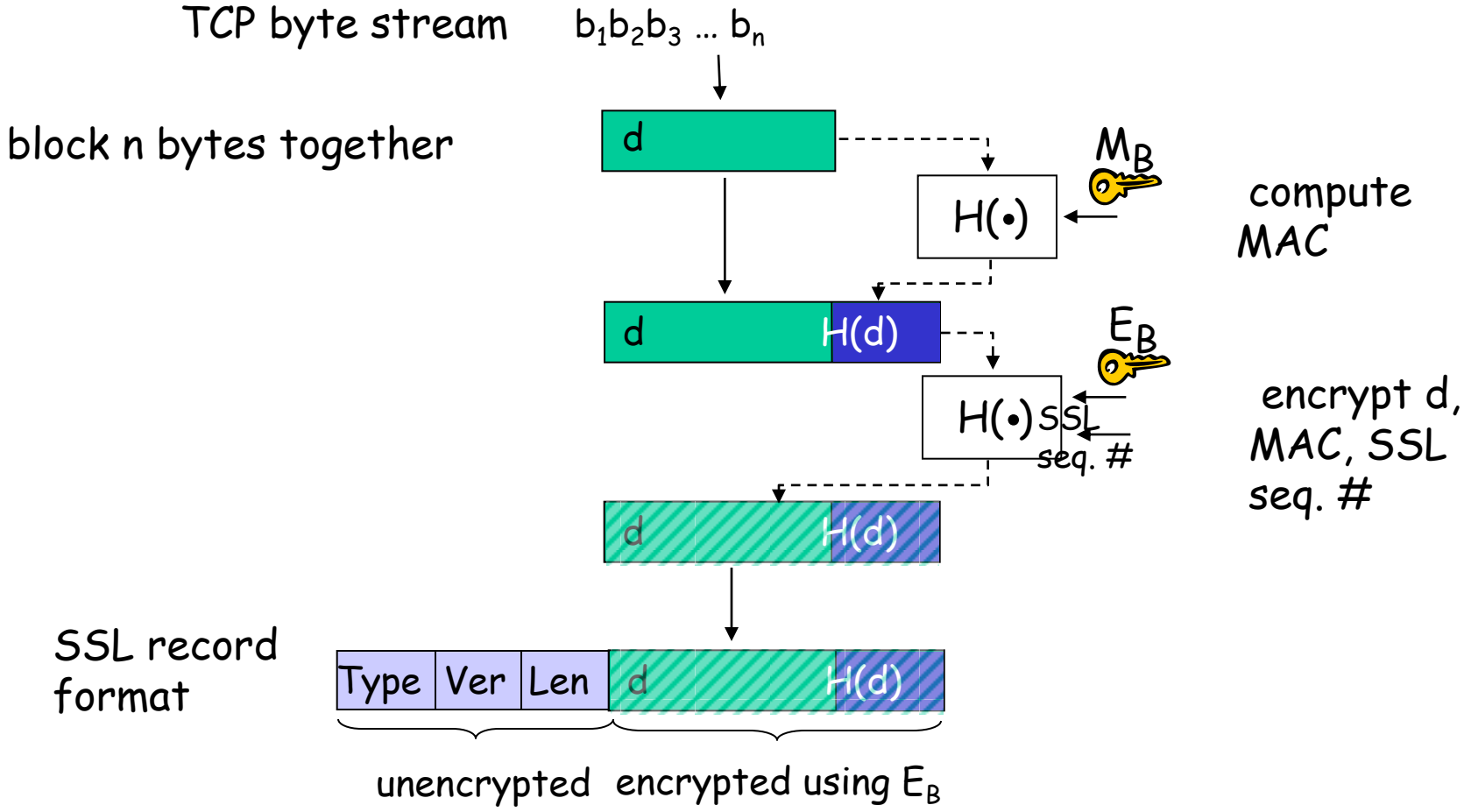
SSL: three phases

2. Key Derivation:

- Alice, Bob use shared secret (MS) to generate 4 keys:
  - E_B: Bob->Alice data encryption key
  - E_A: Alice->Bob data encryption key
  - M_B: Bob->Alice MAC key
  - M_A: Alice->Bob MAC key
- encryption and MAC algorithms negotiable between Bob, Alice
- why 4 keys?
  - 2 encryption keys will be used to encrypt data
  - 2 MAC keys will be used to verify the integrity of the data

SSL: three phases

3. Data transfer



Roadmap

1. What is network security?
2. Principles of cryptography
3. Message integrity
4. End point authentication
5. Securing e-mail
6. Securing TCP connections: SSL
7. Network layer security: IPsec
8. Operational security: firewalls and IDS

IPsec: Network Layer Security

- network-layer secrecy:
  - sending host encrypts the data in IP datagram
  - TCP and UDP segments; ICMP and SNMP messages.
- network-layer authentication
  - destination host can authenticate source IP address (prevent IP address spoofing)
- two principal protocols:
  - authentication header (AH) protocol
  - encapsulation security payload (ESP) protocol

- for both AH and ESP, source, destination handshake:
  - create network-layer logical channel called a security association (SA)
- each SA unidirectional.
- uniquely determined by:
  - security protocol (AH or ESP)
  - source IP address
  - 32-bit connection ID

Authentication Header (AH) Protocol

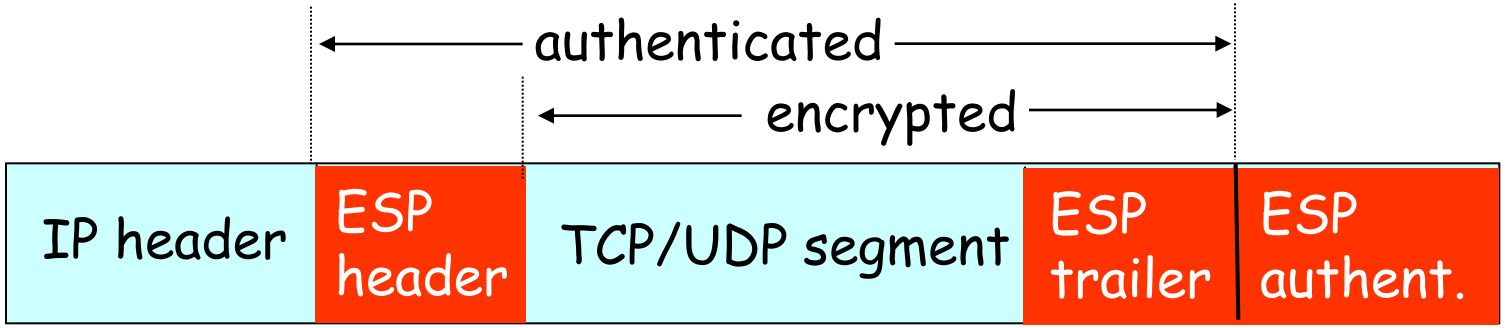
- provides source authentication, data integrity, no confidentiality
- AH header inserted between IP header, data field.
- protocol field: 51
- intermediate routers process datagrams as usual

AH header includes:

- connection identifier
- authentication data: source- signed message digest calculated over original IP datagram.
- next header field: specifies type of data (e.g., TCP, UDP, ICMP)

ESP Protocol

- provides secrecy, host authentication, data integrity.
- data, ESP trailer encrypted.
- next header field is in ESP trailer.
- ESP authentication field is similar to AH authentication field.
- Protocol = 50.



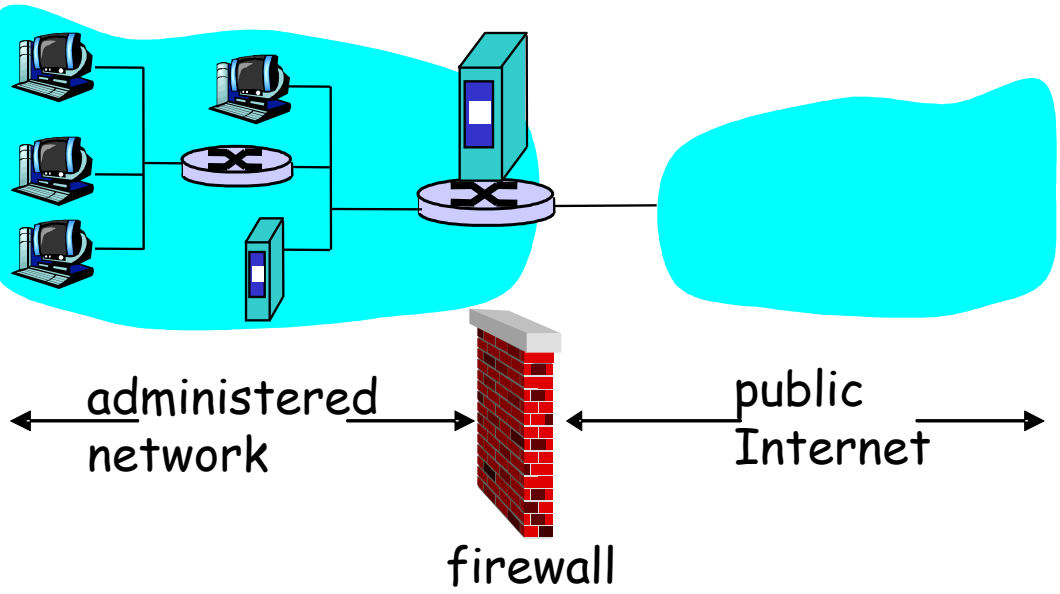
Roadmap

1. What is network security?
2. Principles of cryptography
3. Message integrity
4. End point authentication
5. Securing e-mail
6. Securing TCP connections: SSL
7. Network layer security: IPsec
8. Operational security: firewalls and IDS

Firewalls

firewall

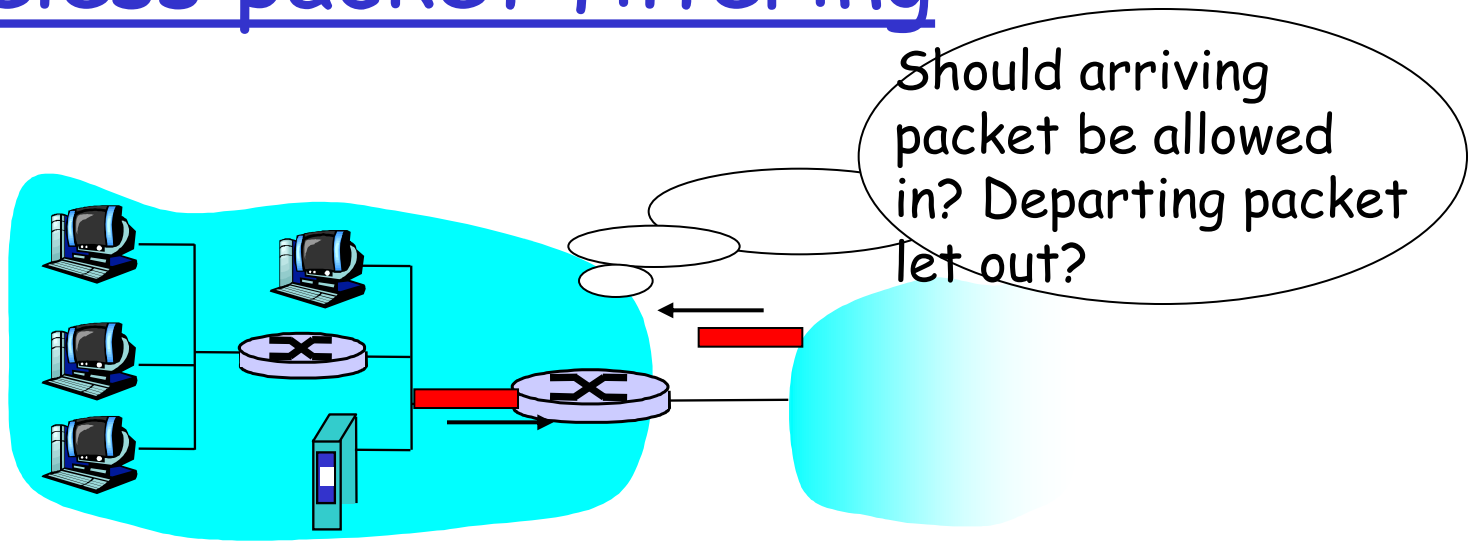
isolates organization's internal net from larger Internet, allowing some packets to pass, blocking others.



Firewalls: Why

- prevent denial of service attacks:
  - SYN flooding: attacker establishes many bogus TCP connections, no resources left for "real" connections
- prevent illegal modification/access of internal data
  - e.g., attacker replaces CIA's homepage with something else
- allow only authorized access to inside network (set of authenticated users/hosts)
- three types of firewalls:
  - stateless packet filters
  - stateful packet filters
  - application gateways

Stateless packet filtering



- internal network connected to Internet via router firewall
- router filters packet-by-packet, decision to forward/drop packet based on:
  - source IP address, destination IP address
  - TCP/UDP source and destination port numbers
  - ICMP message type
  - TCP SYN and ACK bits

Stateless packet filtering: example

- example 1: block incoming and outgoing datagrams with IP protocol field = 17 and with either source or dest port = 23.
  - all incoming, outgoing UDP flows and telnet connections are blocked.
- example 2: Block inbound TCP segments with ACK=0.
  - prevents external clients from making TCP connections with internal clients, but allows internal clients to connect to outside.

Stateless packet filtering: more examples

Policy	Firewall Setting
No outside Web access.	Drop all outgoing packets to any IP address, port 80
No incoming TCP connections, except those for institution's public Web server only.	Drop all incoming TCP SYN packets to any IP except 130.207.244.203, port 80
Prevent Web-radios from eating up the available bandwidth.	Drop all incoming UDP packets - except DNS and router broadcasts.
Prevent your network from being used for a smurf DoS attack.	Drop all ICMP packets going to a "broadcast" address (eg 130.207.255.255).
Prevent your network from being tracerouted	Drop all outgoing ICMP TTL expired traffic

Access Control Lists

- *ACL:* table of rules, applied top to bottom to incoming packets: (action, condition) pairs

action	source address	dest address	protocol	source port	dest port	flag bit
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	any
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	---
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	----
deny	all	all	all	all	all	all

Stateful packet filtering

- stateless packet filter: heavy handed tool
  - admits packets that "make no sense," e.g., dest port = 80, ACK bit set, even though no TCP connection established:

action	source address	dest address	protocol	source port	dest port	flag bit
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK

- *stateful packet filter:* track status of every TCP connection
  - track connection setup (SYN), teardown (FIN): can determine whether incoming, outgoing packets "makes sense"
  - timeout inactive connections at firewall: no longer admit packets

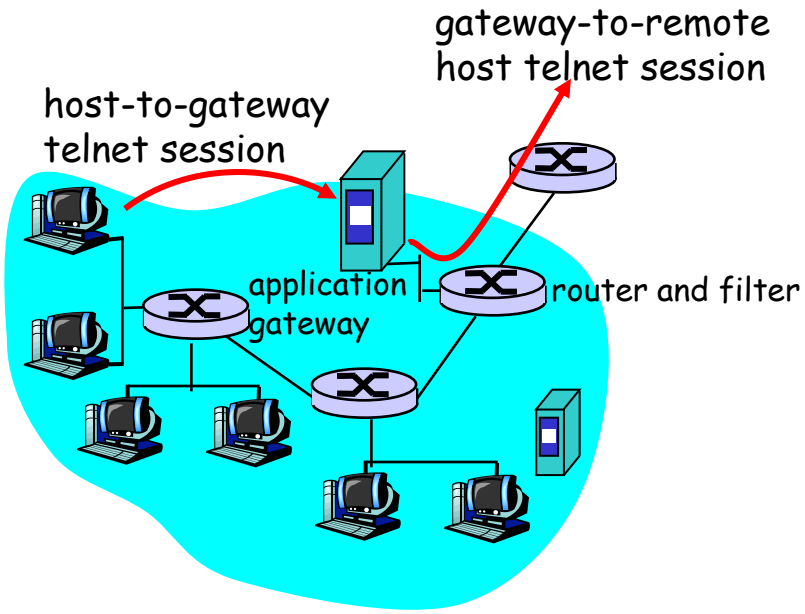
Stateful packet filtering

- ACL augmented to indicate need to check connection state table before admitting packet

action	source address	dest address	proto	source port	dest port	flag bit	check conxion
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	any	
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK	X
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	---	
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	----	X
deny	all	all	all	all	all	all	

Application gateways

- filters packets on application data as well as on IP/TCP/UDP fields.
- example: allow select internal users to telnet outside.



1. require all telnet users to telnet through gateway.
2. for authorized users, gateway sets up telnet connection to dest host. Gateway relays data between 2 connections
3. router filter blocks all telnet connections not originating from gateway.

Limitations of firewalls and gateways

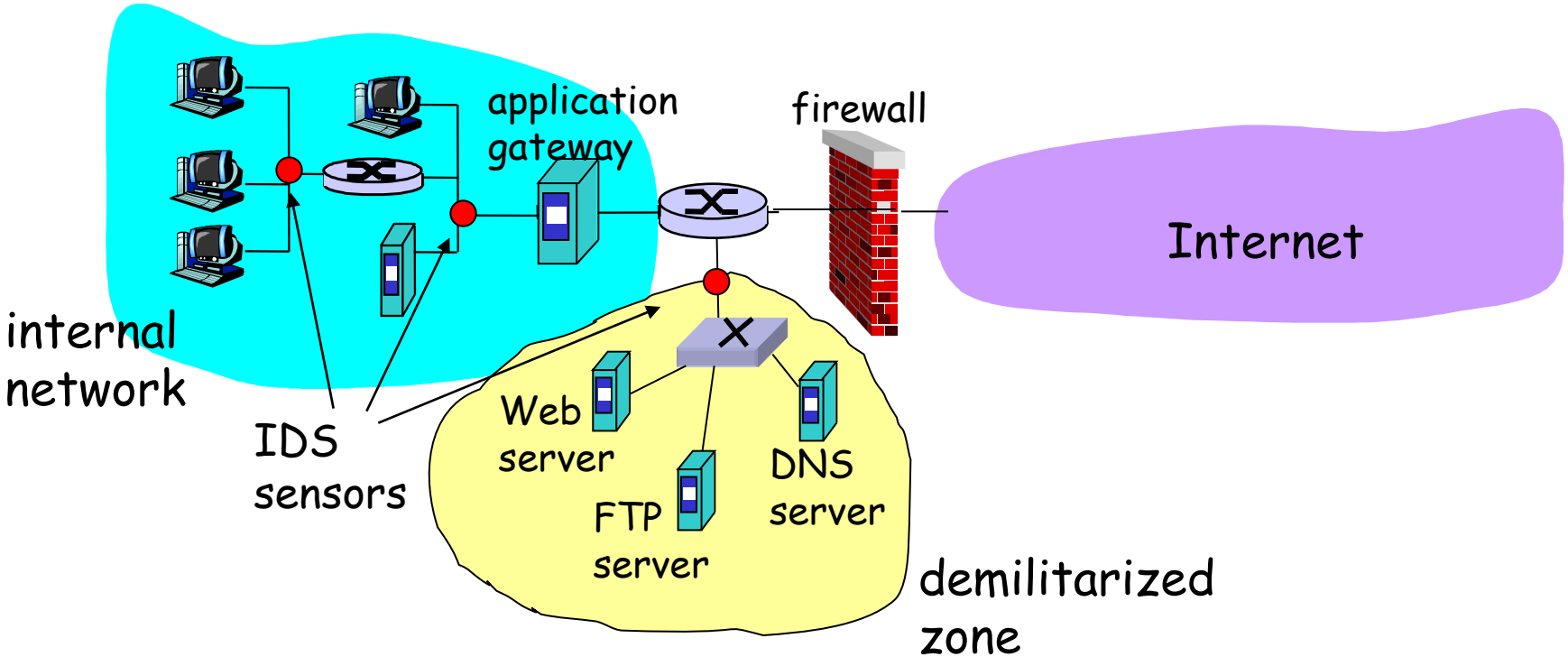
- IP spoofing: router can't know if data "really" comes from claimed source
- if multiple app's. need special treatment, each has own app. gateway.
- client software must know how to contact gateway.
  - e.g., must set IP address of proxy in Web browser
- filters often use all or nothing policy for UDP.
- tradeoff: degree of communication with outside world, level of security
- many highly protected sites still suffer from attacks.

Intrusion detection systems

- packet filtering:
  - operates on TCP/IP headers only
  - no correlation check among sessions
- *IDS: intrusion detection system*
  - *deep packet inspection:* look at packet contents (e.g., check character strings in packet against database of known virus, attack strings)
  - examine correlation among multiple packets
    - port scanning
    - network mapping
    - DoS attack

Intrusion detection systems

- multiple IDSs: different types of checking at different locations



Network Security (summary)

Basic techniques……

- cryptography (symmetric and public)
- message integrity
- end-point authentication

…. used in many different security scenarios

- secure email
- secure transport (SSL)
- IP sec

Operational Security: firewalls and IDS

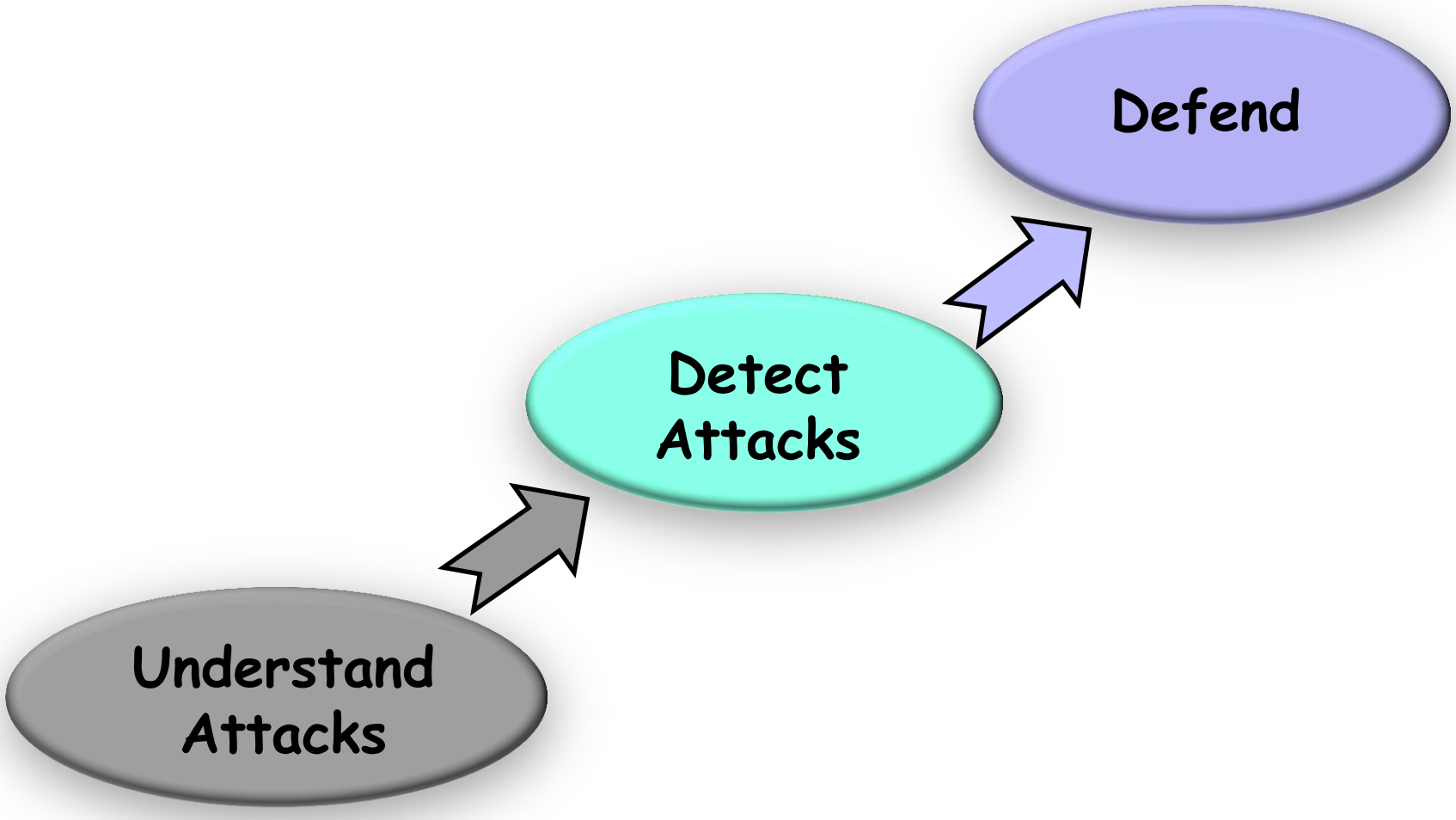
Outline

- General topics in network security
- Wireless Security
- IoT Security

Why is Security More of a Concern in Wireless?

- No inherent physical protection
  - physical connections between devices are replaced by logical associations
  - sending and receiving messages do not need physical access to the network infrastructure (cables, hubs, routers, etc.)
- Broadcast communications
  - wireless usually means radio, which has a broadcast nature
  - transmissions can be overheard by anyone in range
  - anyone can generate transmissions, which will be received by other devices in range
  - which will interfere with other nearby transmissions and may prevent their correct reception (jamming)
- Eavesdropping is easy
- Injecting bogus messages into the network is easy
- Replaying previously recorded messages is easy
- Illegitimate access to the network and its services is easy
- Denial of service is easily achieved by jamming

Overview



Attacking Wireless Networks

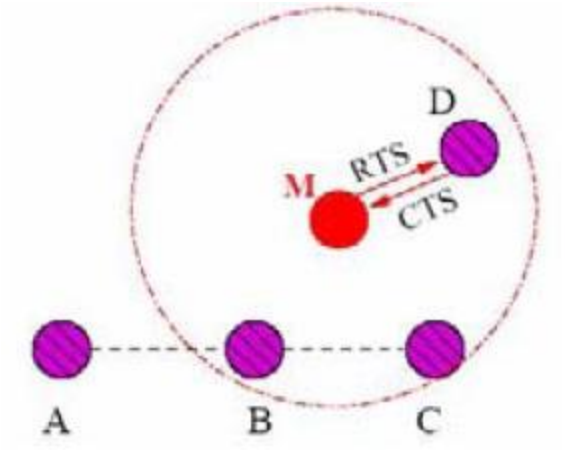
Attack 1: CSMA Selfish Behaviors

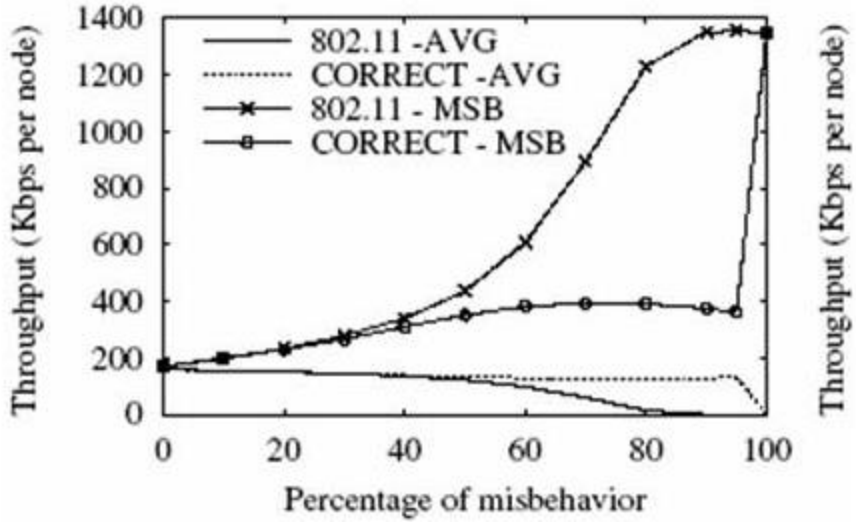
- Carrier sense: When a node wishes to transmit a packet, it first waits until the channel is idle
- Backoff Interval: used to reduce collision probability
- When transmitting a packet, choose a backoff interval in the range [0,cw]: cw is contention window
- Count down the backoff interval when medium is idle
  - Count-down is suspended if medium becomes busy
- When backoff interval reaches 0, transmit
- IEEE 802.11 DCF: contention window cw is chosen dynamically depending on collision occurrence

Binary Exponential Backoff

- When a node faced a transmission failure, it increases the contention window
  - *cw* is doubled (up to an upper bound)
- When a node successfully completes a data transfer, it restores *cw* to CW_{min}
- *cw* follows a sawtooth curve

Attack 1: CSMA Selfish Behaviors

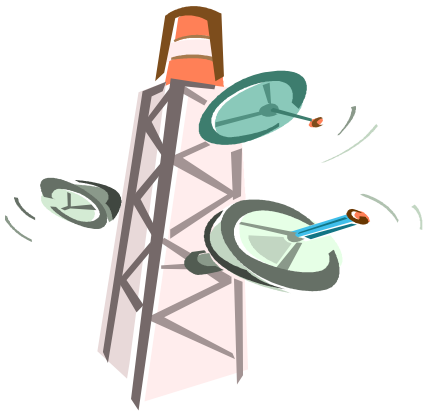




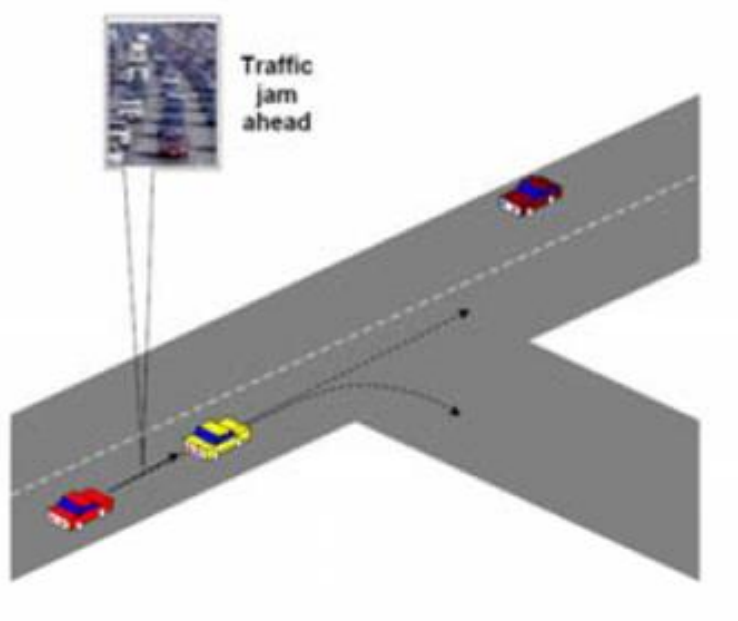
- Use smaller backoff window
  - Transmit with you should not
    - Attacker gets more bandwidth
  - Cause collisions to others

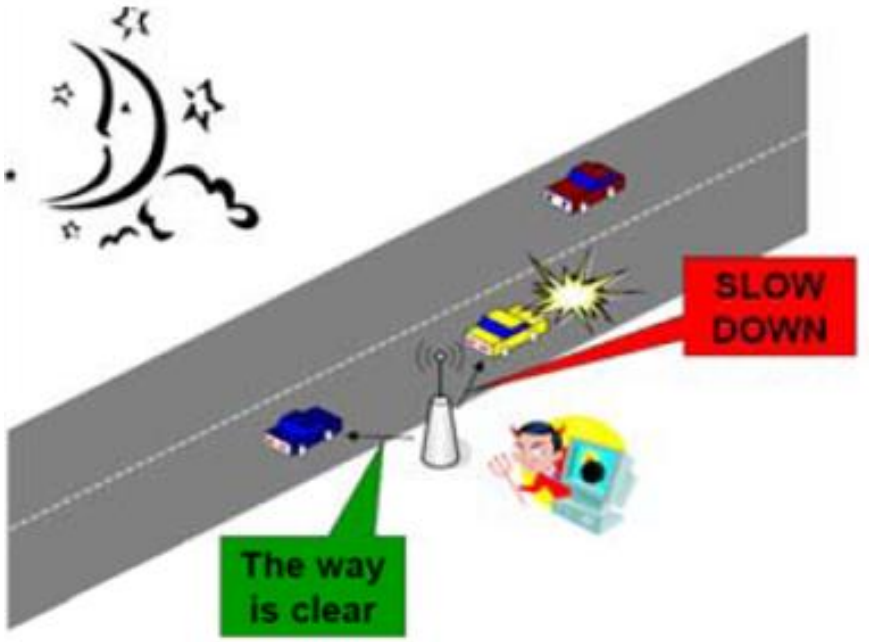
Attack 2: Jamming

jammer



Attack 3: Injecting Bogus Information

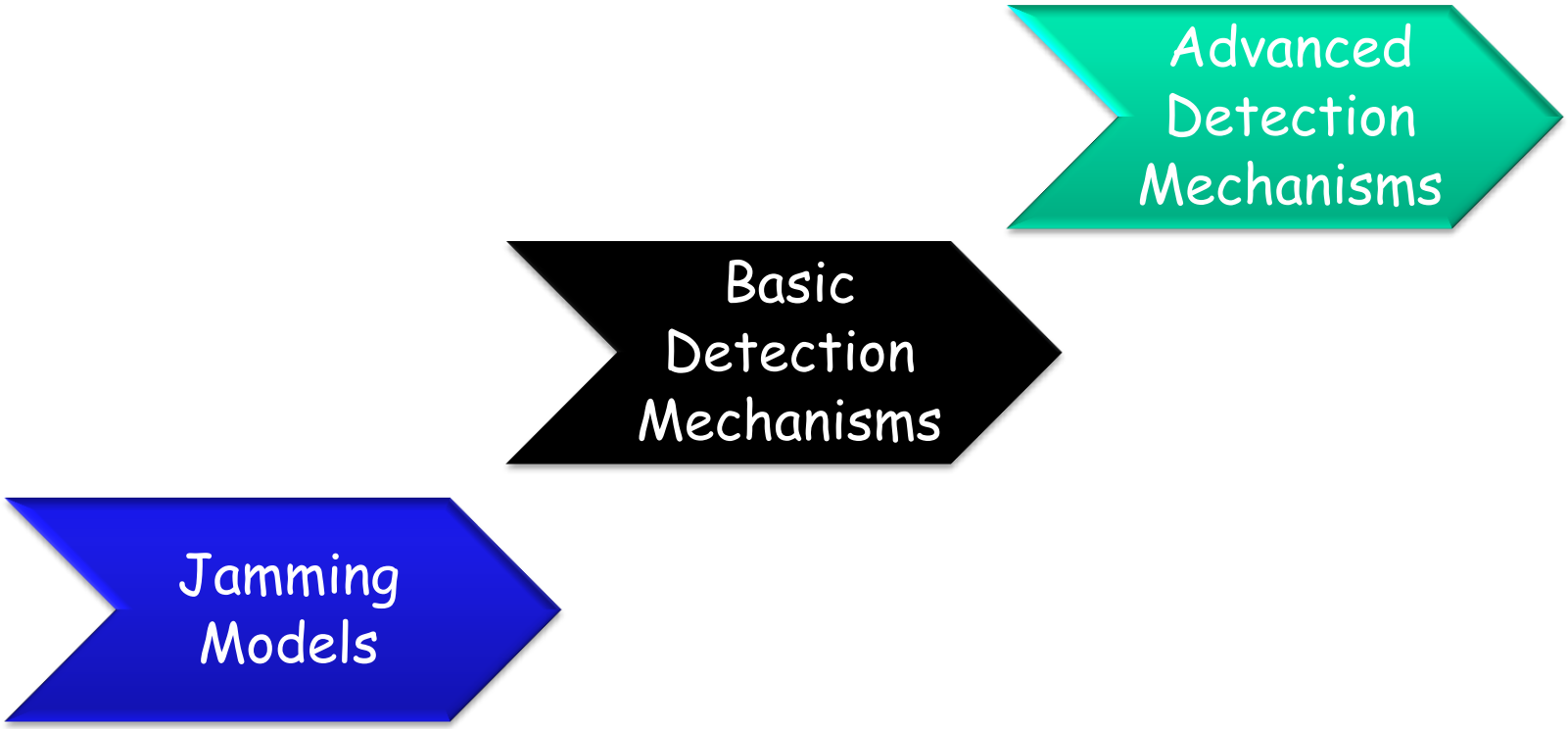




Introduction

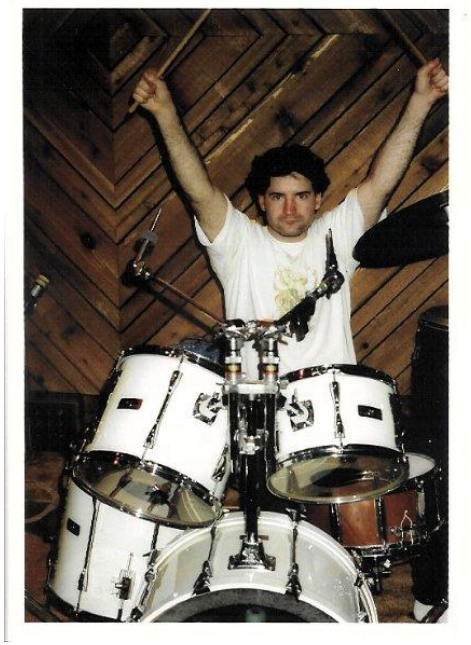
- Traditionally, Denial of Service (DoS) attacks involve filling receiving buffers and/or bringing down servers
- In the wireless domain, DoS is more fundamentally linked with the medium
  - MAC misbehavior or
  - Preventing nodes from even communicating (*i.e., jamming)*

Roadmap


Advanced Detection Mechanisms
Basic Detection Mechanisms
Jamming Models

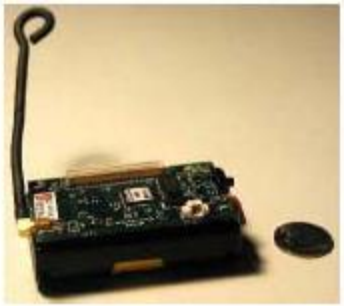
What is a Jammer?

- A jammer is purposefully trying to interfere with the physical transmit/receive



Jammers -- Hardware

- Cell phone jammer unit
  - Block every cellular phone!!!
- Signal generator
- Conventional devices

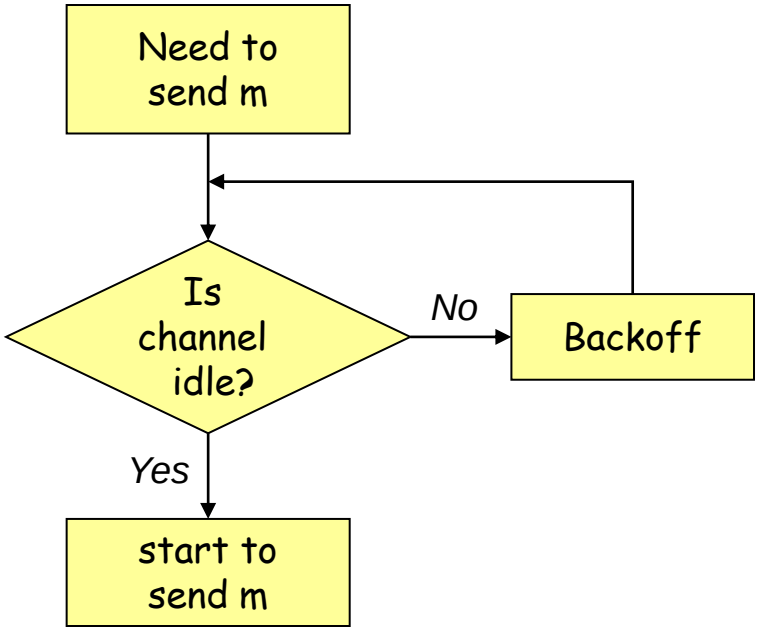


Goal of Jammer

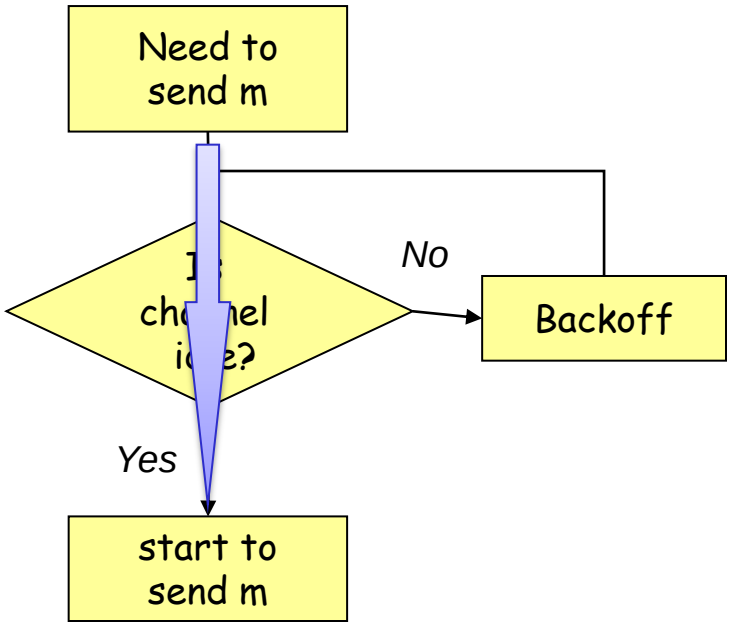
- Interference
  - Prevent a sender from sensing out packets
  - Prevent a receiver from receiving legitimate packets
- How to measure their effectiveness:
  - Packet Send Ratio (PSR):
    - Ratio of actual # of packets sent out versus # of packets intended
  - Packet Delivery Ratio (PDR):
    - Ratio of # of successfully delivered packets versus # of packets sent out
    - Measured at sender [ACKs] or receiver [CRC check]

Jammer Attack Models

Normal MAC protocol:



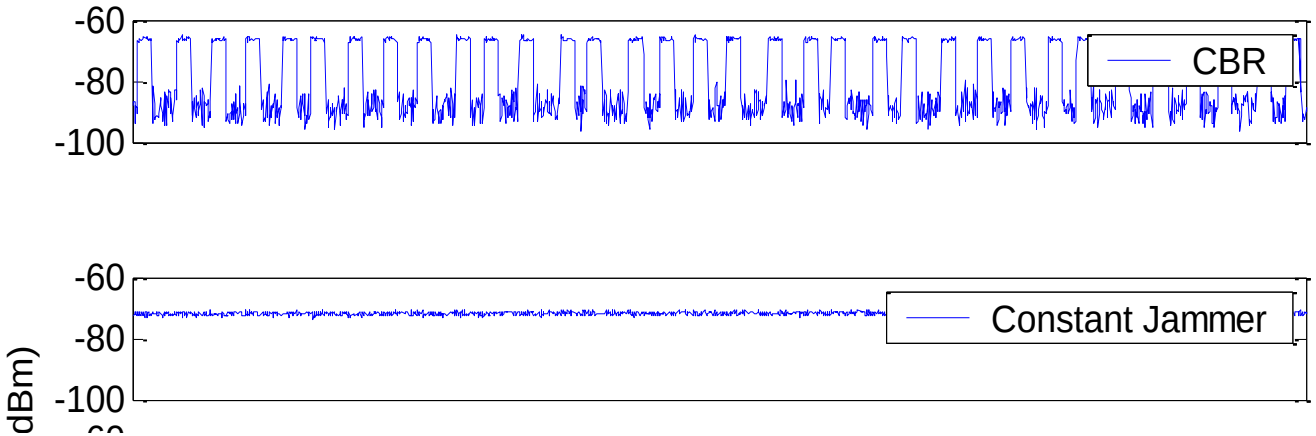
Jammer:



Jamming Models

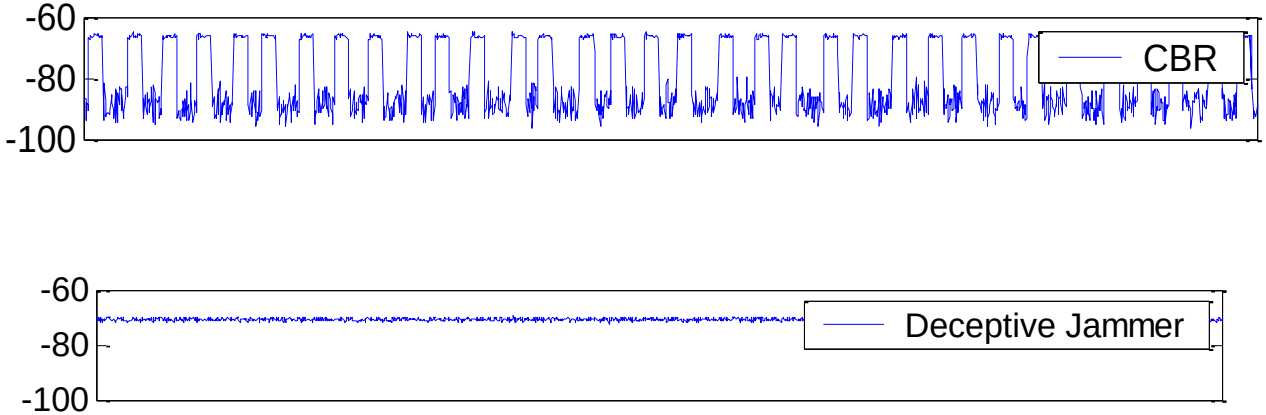
- Constant Jammer
  - Continuously emits radio signal (random bits, no MAC-etiquette)
- Deceptive Jammer
  - Continuously sends regular packets (preamble bits) without gaps in transmission
  - Targeting sending
- Random Jammer
  - Alternates between sleeping and jamming states.
  - Takes energy conservation into consideration
- Reactive Jammer:
  - Reacts to a sent message
  - Targeting reception;
  - Harder to detect

Constant Jammer



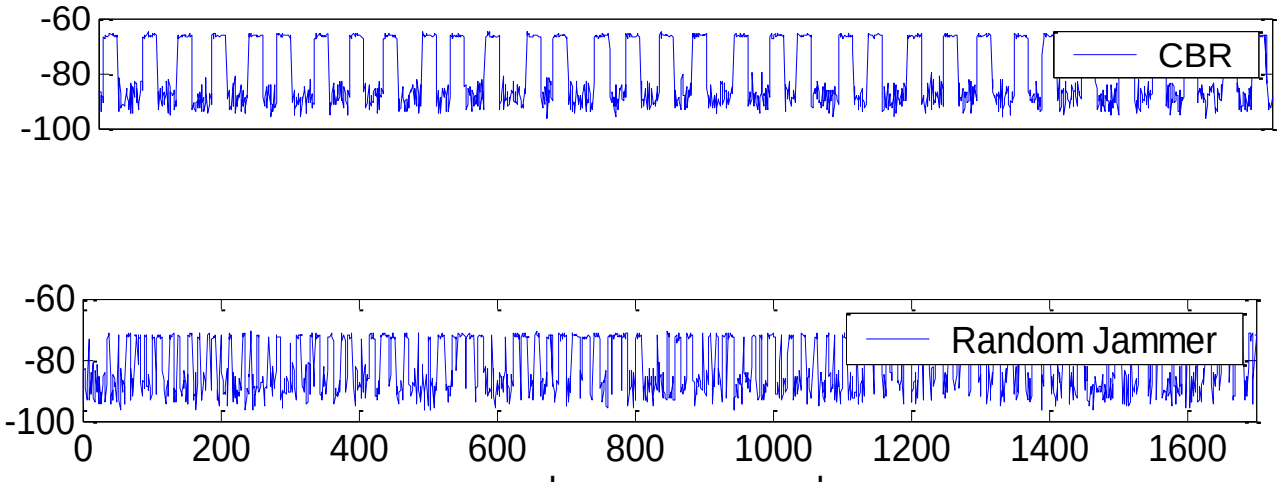
- Constant Jammer - continually emits a radio signal (noise). The device will not wait for the channel to be idle before transmitting. Can disrupt even signal strength comparison protocols .

Deceptive Jammer



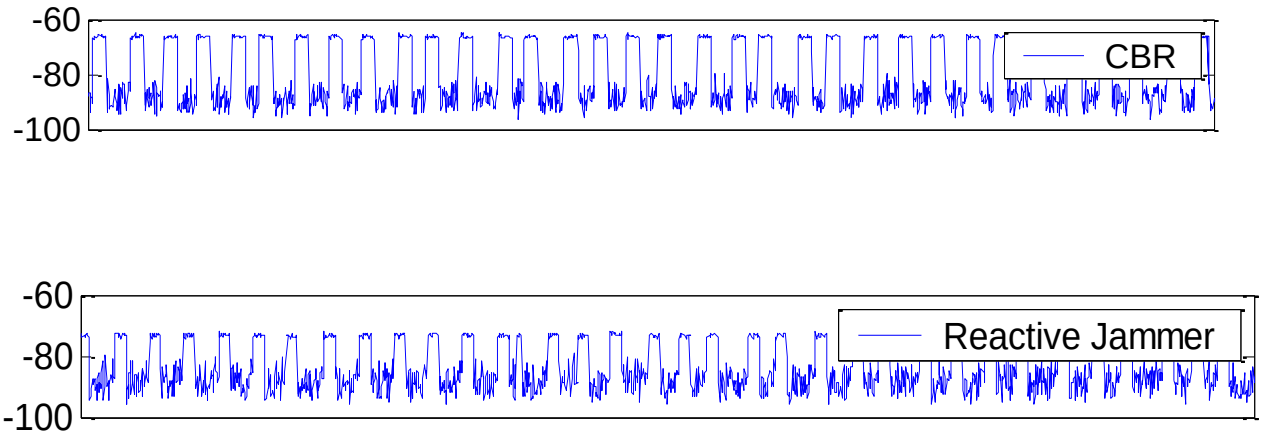
- Deceptive Jammer - constantly injects regular packets with no gap between packets. A normal device will remain in the receive state and cannot switch to the send state because of the constant stream of incoming packets.

Random Jammer



- Random Jammer - alternates between sleeping and jamming. Can act as constant or deceptive when jamming. Takes energy conservation into consideration.

Reactive Jammer



- Reactive Jammer - other three are active this is not. It stays quiet until there is activity on the channel. This targets the reception of a message. This style does not conserve energy however it may be harder to detect.

Basic Jamming Detection

What attributes will help us detect jamming?

- Signal strength (PHY-layer detection)
- Carrier sense (MAC layer detection)
- Packet Delivery Ratio
  - Detects all jamming models
  - Differentiates jamming from congestion
  - Cannot differentiate jamming from node failure, battery loss, departure, etc.

Detection 1: Analyzing Signal Strength

How can we use Signal Strength to detect Jamming?

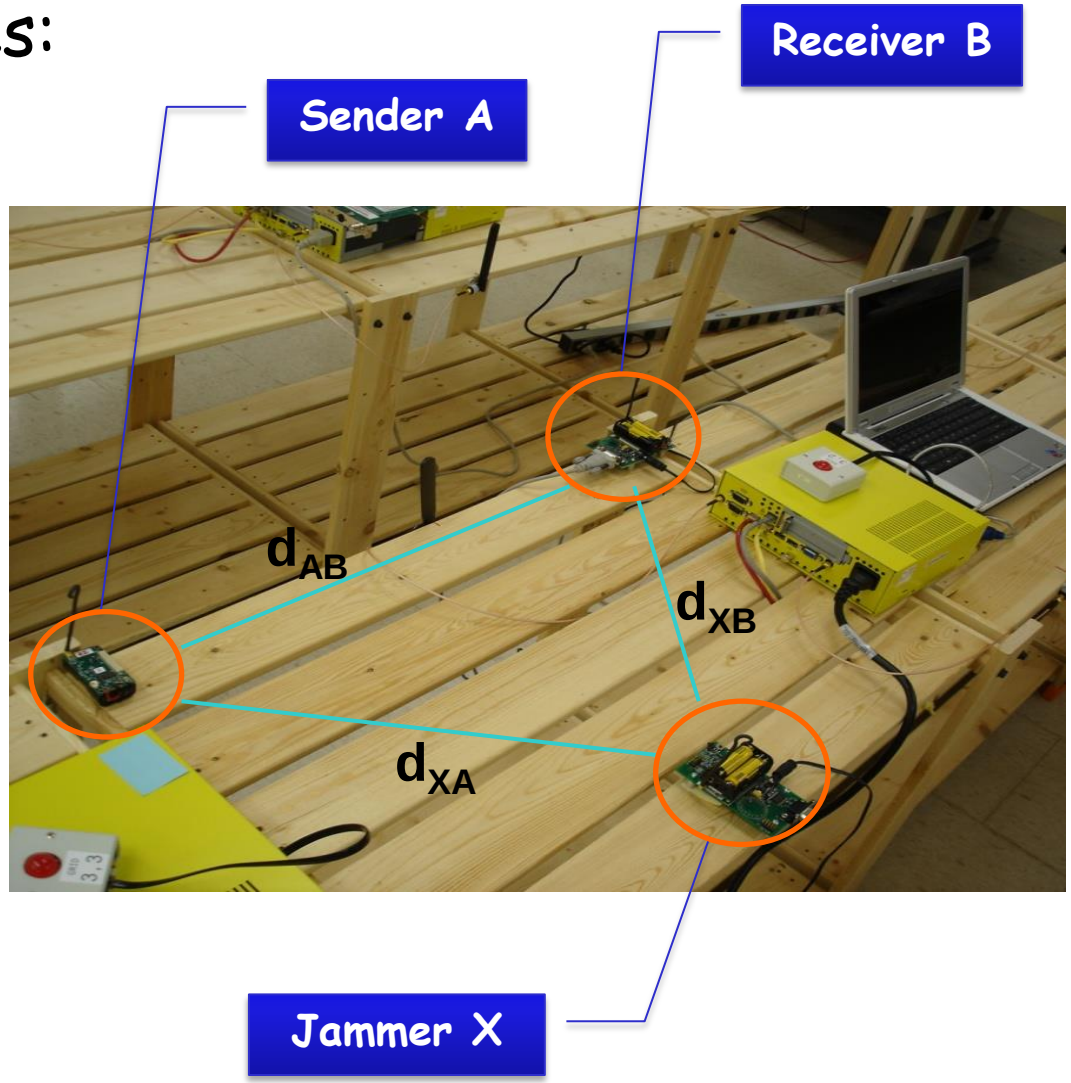
- Signal strength distribution may be affected by the presence of a jammer
- Each device should gather its own statistics to make its own decisions on the possibility of jamming
- Establish a base line or build a statistical model of normal energy levels prior to jamming of noise levels….But how??

Two Methods for Signal Strength

1. Basic Average and Energy Detection
   - We can extract two statistics from this reading, the average signal strength and the energy for detection over a period of time
2. Signal Strength Spectral Discrimination
   - A method that employs higher order crossings (HOC) to calculate the differences between samples
   - This method is practical to implement on resource constrained wireless devices, such as sensor nodes

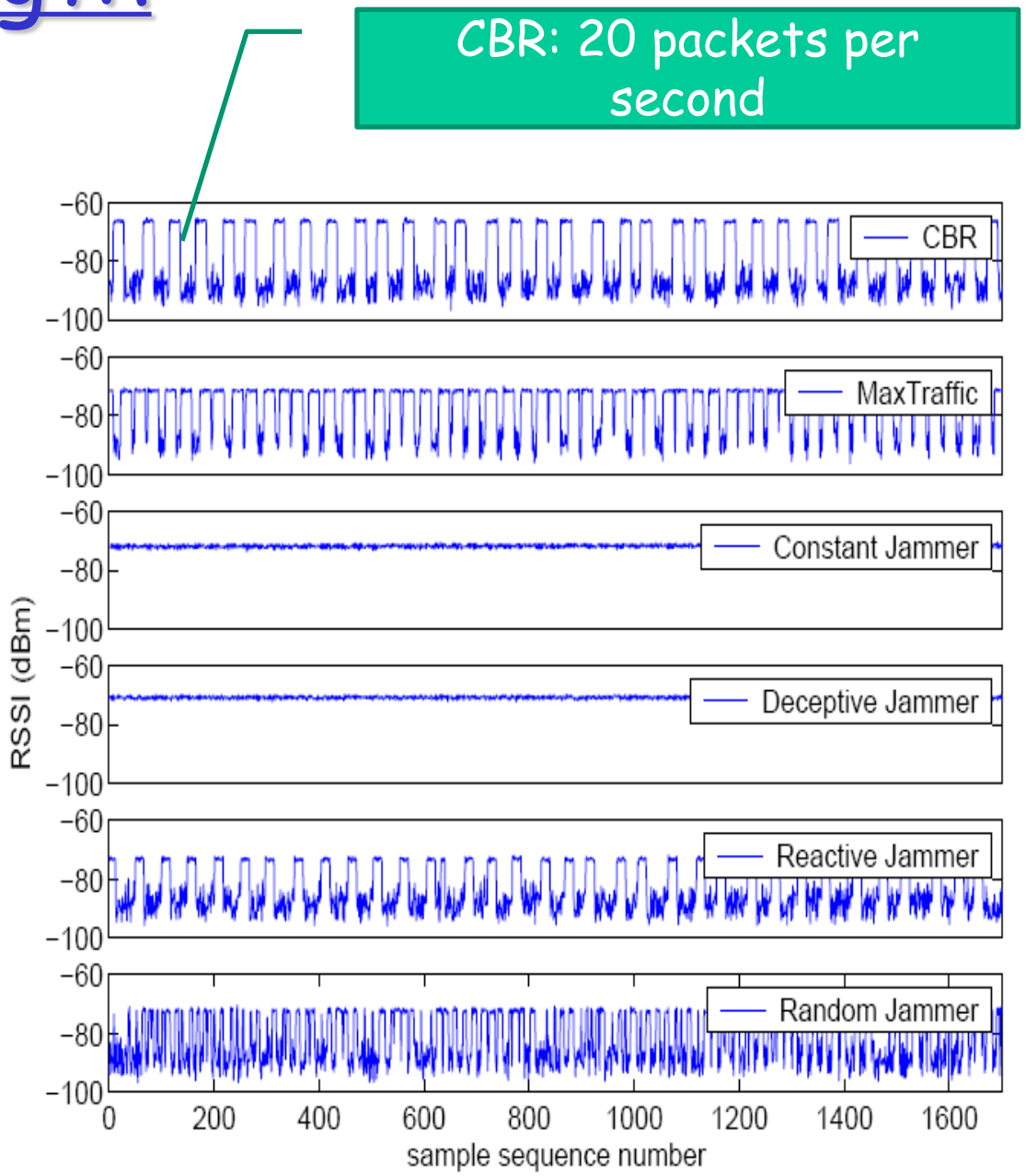
Experiment Setup

- Involving three parties:
  - Normal nodes:
    - Sender A
    - Receiver B
  - Jammer X
- Parameters
  - Four jammers model
  - Distance
    - Let $d_{XB} = d_{XA}$
    - Fix d_{AB} at 30 inches
  - Power
    - $P_A = P_B = P_X = -4dBm$



Signal Strength

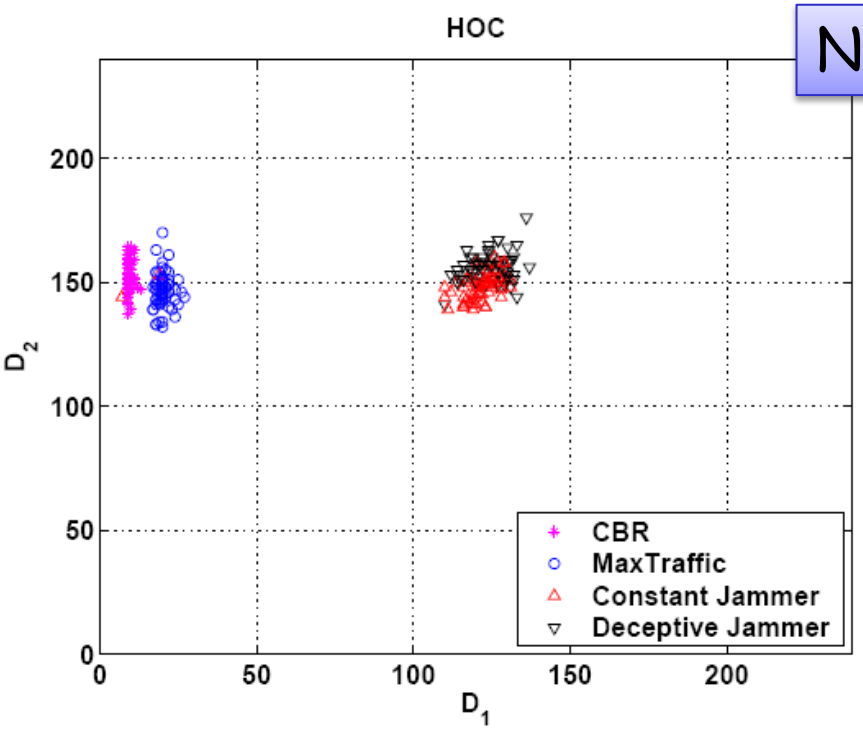
- The average values for the constant jammer and the MaxTraffic source are roughly equal
- The constant jammer and deceptive jammer have roughly the same average values
- The signal strength average from a CBR source does not differ much from the reactive jammer scenario
- These results suggest that we may not be able to use simple statistics such as average signal strength to identify jamming

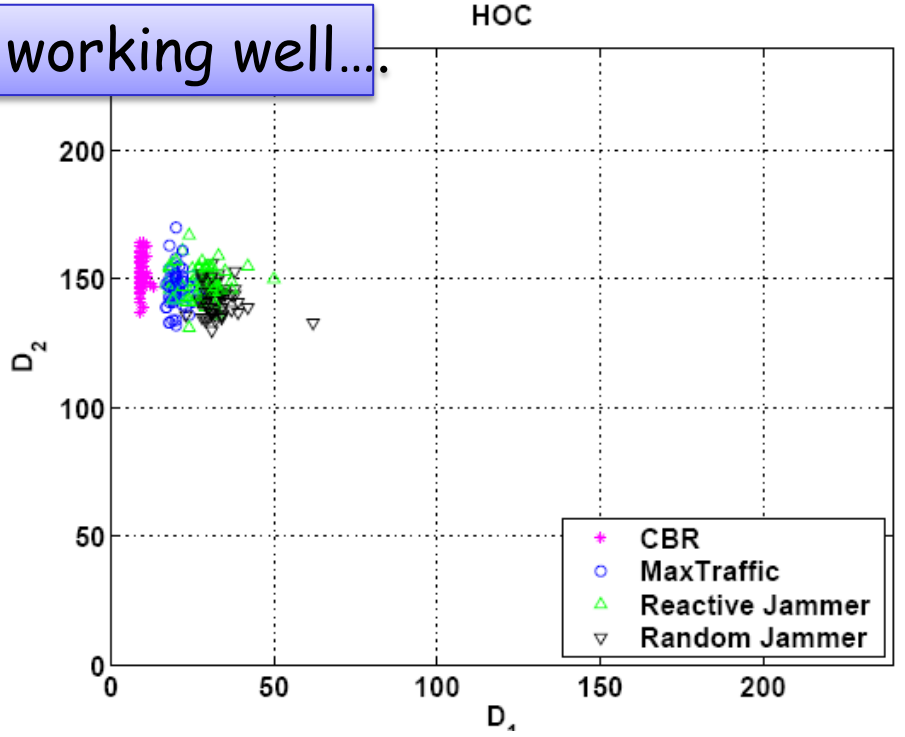


Signal-Strength: Higher Order Crossing

- We can not distinguish the reactive or random jammer from normal traffic
- A reactive or random jammer will alternate between busy and idle in the same way as normal traffic behaves
- HOC will work for some jammer scenarios but are not powerful enough to detect all jammer scenarios

Not working well….

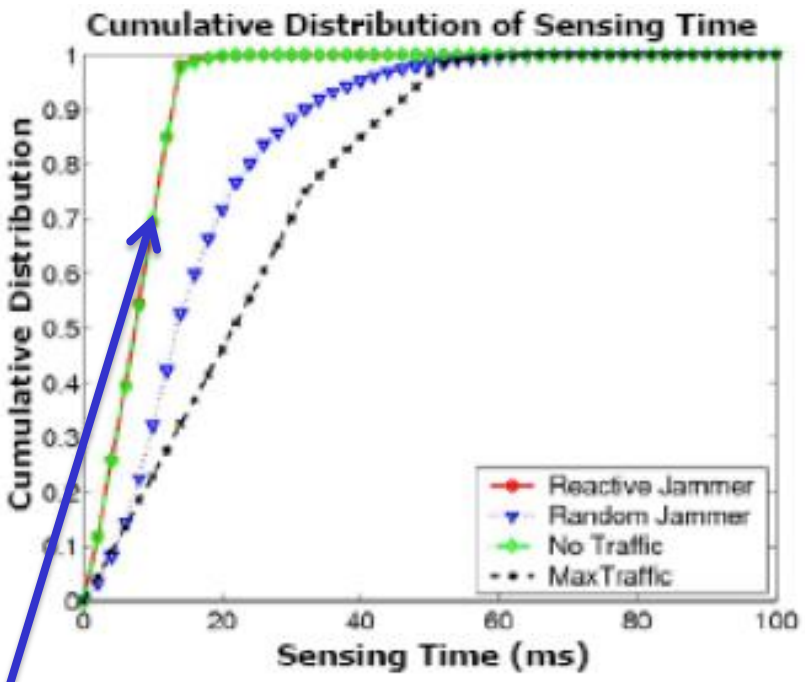


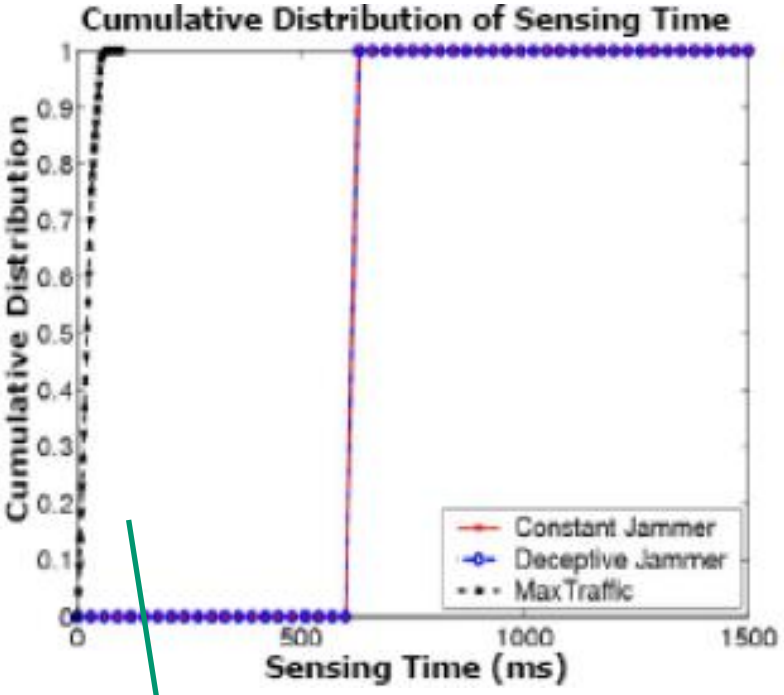


Detection 2: Analyzing Carrier Sensing Time

- A jammer can prevent a legitimate source from sending out packets ← channel might appear constantly busy to the source
- Keep track of the amount of time it spends waiting for the channel to become idle (carrier sensing time)
  - Compare it with the sensing time during normal traffic operations to determine whether it is jammed
  - Only true if MAC protocol employs a fixed signal strength threshold to determine whether channel is busy
- Determine when large sensing times are results of jamming by setting a threshold
- Threshold set conservatively to reduce false positive

Detection 2: Analyzing Carrier Sensing Time



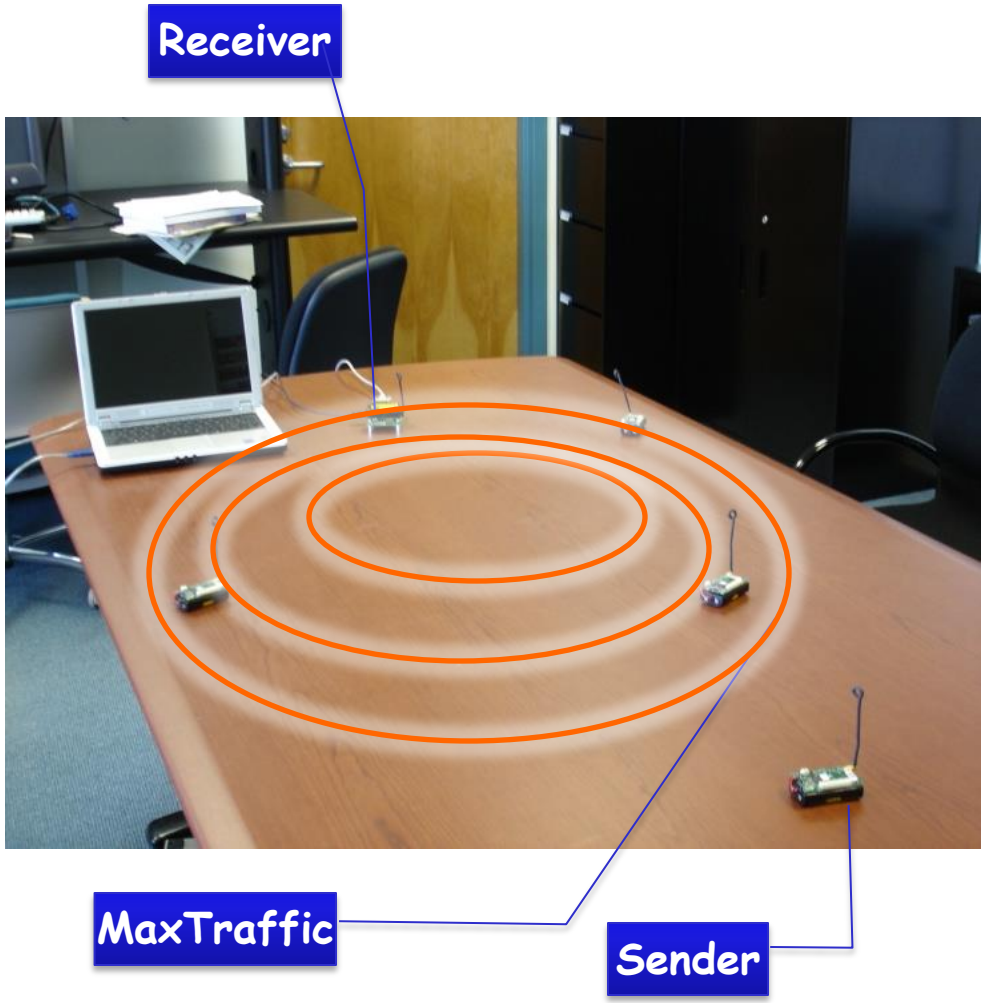


Hard to detect a reactive jammer

It detects the Constant and Deceptive Jammer

Detection 3: Analyzing Packet Delivery Ratio

- How much PDR degradation can be caused by non-jamming, normal network dynamics, such as congestion? (PDR 78%)
- A jammer causes the PDR drop significantly, almost to 100%
- A simple threshold based on PDR is a powerful statistic to determine Jamming vs. congestion
- PDR can not differentiate non-aggressive jamming attacks from poor channel quality…



Basic Statistics Summary

- Both Signal Strength and Carrier Sensing time can only detect the constant and deceptive jammer
- Neither of these two statistics is effective in detecting the random or the reactive jammer
- PDR is a powerful statistic to determine Jamming vs. congestion
  - It can not account for all network dynamics

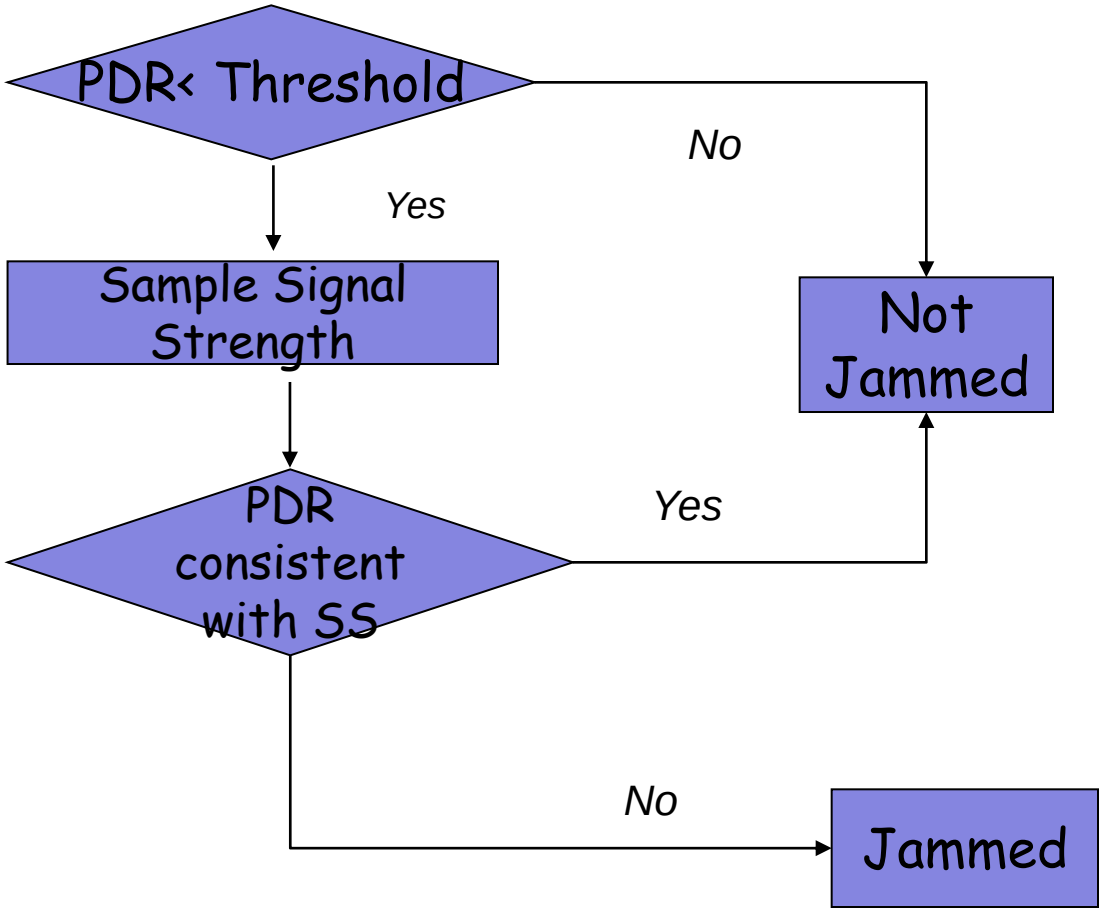
Solution: Consistency Checks

- PDR is relatively good
  - Normal scenario:
    - High signal strength → high PDR
    - Low signal strength → low PDR
  - Low PDR in real life
    - Poor channel quality
    - Jamming attacks → high signal strength
- Consistency check
  - Look at transmissions from neighbors
  - If at least one neighbor has high PDR
  - If all have low PDR → check signal strength → high → I am being jammed!

Location-Based Consistency Check

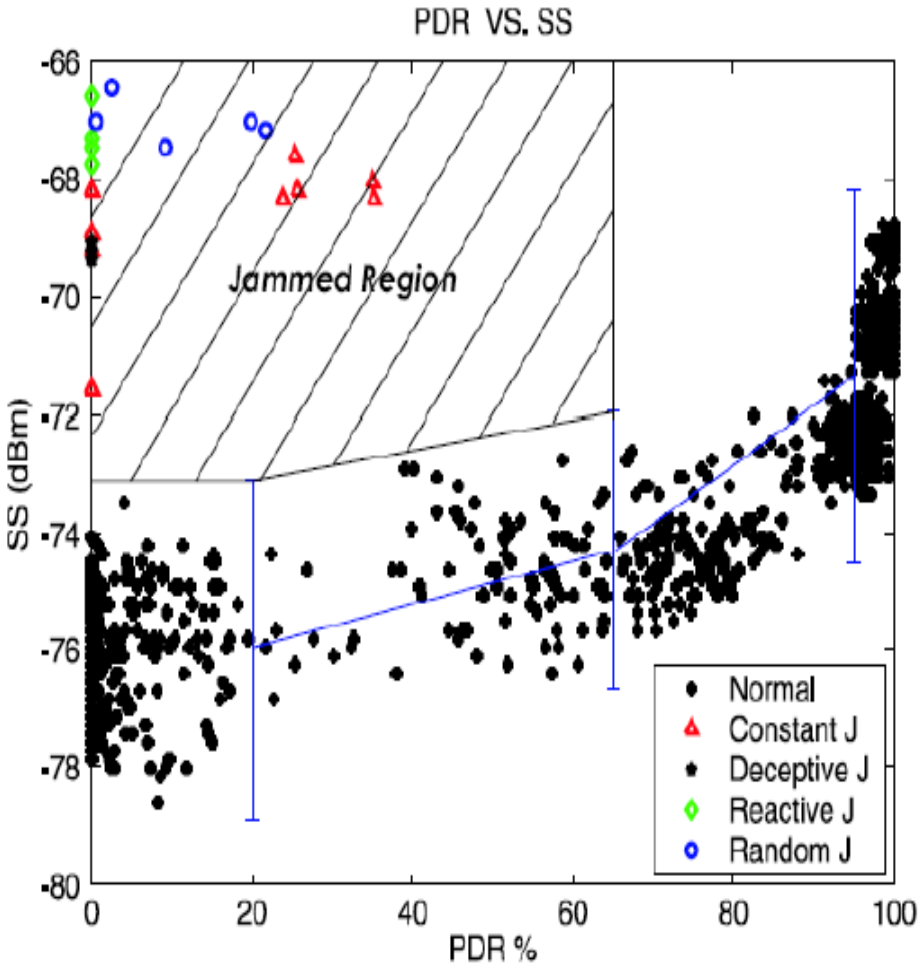
- Concept:
  - Close neighbor nodes → high PDR
  - Far neighbor nodes → lower PDR
- If all nearby neighbors exhibit low PDR
  → jammed!

PRD/Signal Strength Consistency



Results

- Observed Normal relationships
  - High signal strength yields a high PDR
  - Low signal strength yields a low PDR
- Jammed scenario: a high signal strength but a low PDR
- The Jammed region has above 99% signal strength confidence intervals and whose PDR is below 65%

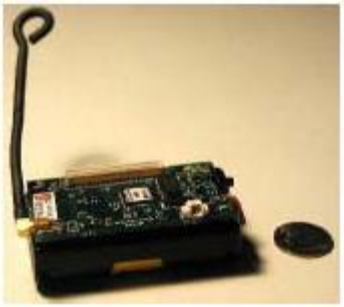


What Happens After Detection??

- This work has identified jamming models and described a means of detection
- Prevention? Reaction?
  - Channel surfing and spatial retreats
  - SSCH

Our Jammers

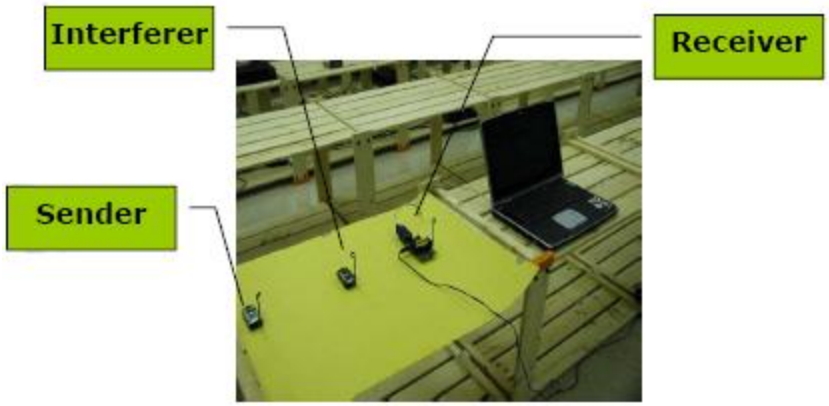
- MAC-layer Jammer
  - Mica2 Motes (UC Berkeley)
    - 8-bit CPU at 4MHz
    - 512KB flash, 4KB RAM
    - 916.7MHz radio
    - OS: TinyOS
  - Disable the CSMA
  - Keep sending out the preamble
- PHY-layer Jammer
  - Waveform Generator
  - Tune frequency to 916.7MHz

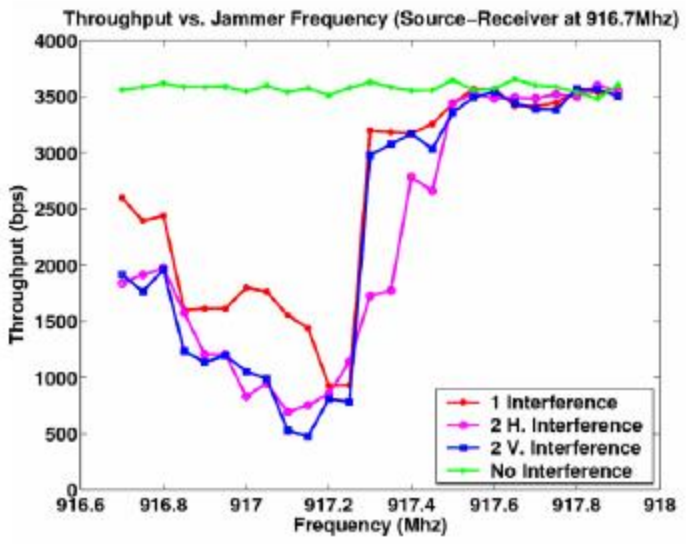


Escaping From Jamming Attacks

- Channel Surfing
  - Utilize frequency hopping if a node detects that it is being jammed it just switches to another channel
    - Inspired by frequency hopping techniques, but operates at the link layer
  - System Issues: Must have ability to choose multiple "orthogonal" channels
    - Practical Issue: PHY specs do not necessarily translate into correct "orthogonal" channels
    - Example: MICA2 Radio recommends: "choose separate channels with a minimum spacing of 150KHz" but…..

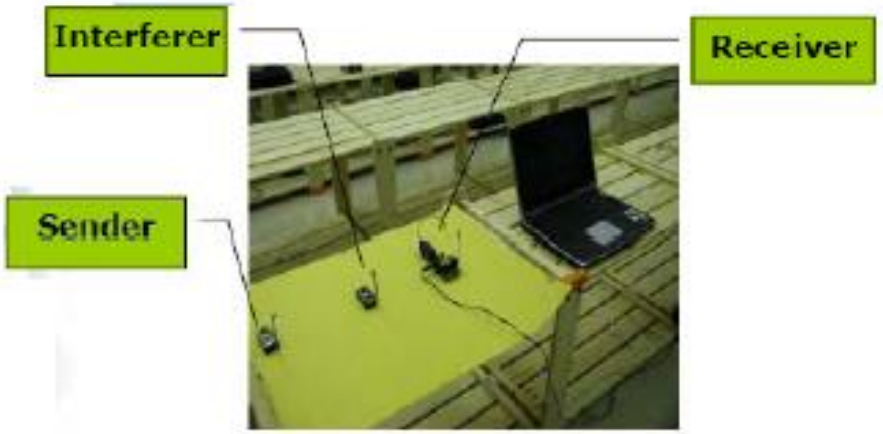
Throughput VS. Channel Assignment

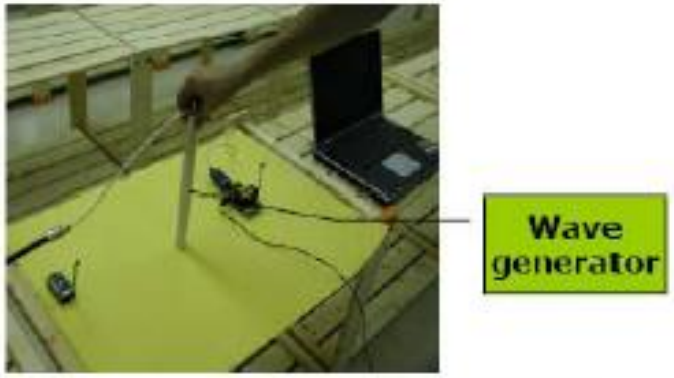


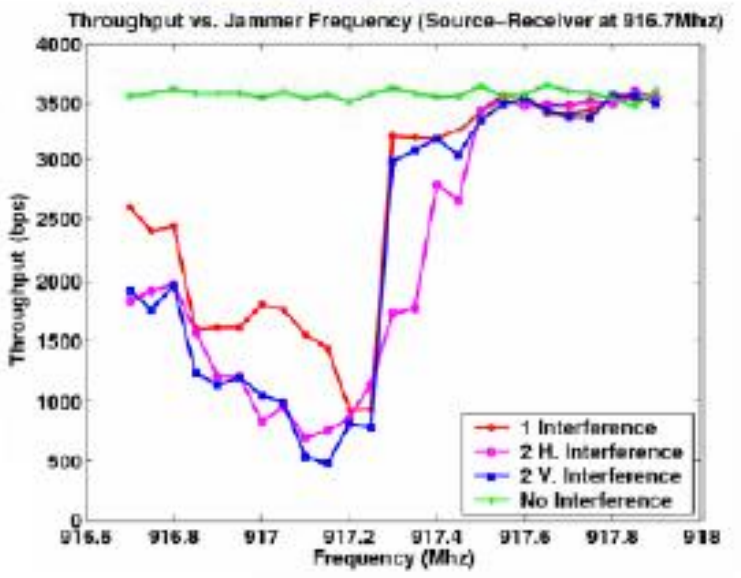


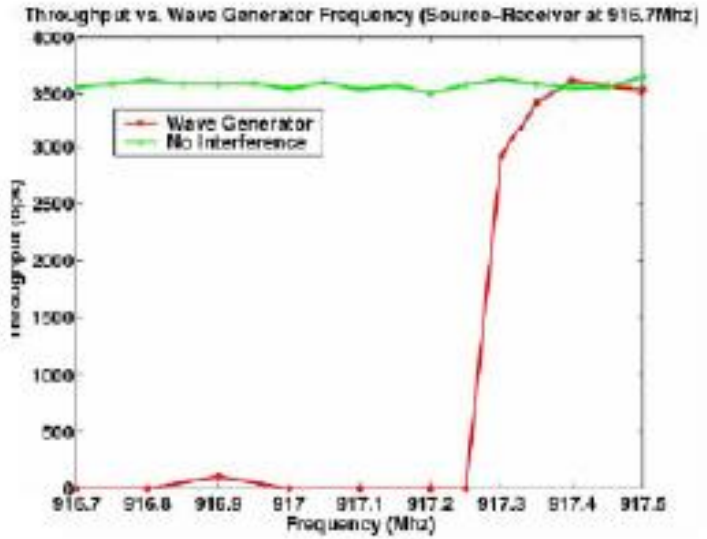
- Sender sends the packet as fast as it can
- Receiver counts the packet and calculates the throughput
- The radio frequency of the sender and receiver was fixed at 916.7MHz
- Increased the interferer's communication frequency by 50kHz each time
- When the Jammer's communication frequency increases to 917.5MHz, there is almost no interference

Is Channel Surfing Feasible??









Escaping From Jamming Attacks

- "Orthogonal" channels
  - The fact is that we need at least 800KHz to escape the interference
  - Therefore, explicit determination of the amount of orthogonal channels is important
- Channel Surfing
  - Target: maximize the delay before the attacker finds the new channel
  - Solution: use a (keyed) pseudo-random channel assignment between nodes

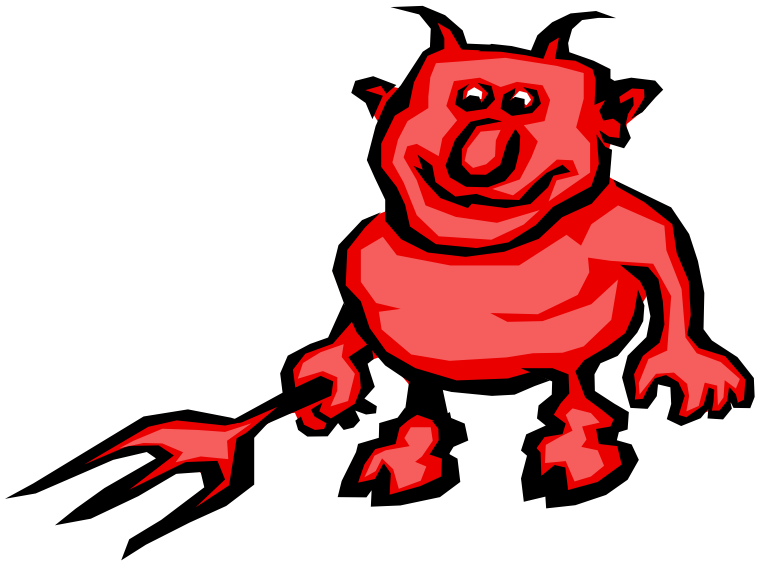
Escaping from Jamming Attacks

- Spatial Retreats:
  - If a node is jammed move spatially (physically) to another location
  - When a node changes location, it needs to move to a new location where it can avoid being jammed but minimize network degradation
  - Sometimes a spatial retreat will cause a network partition
- Two different strategies to defend against jamming
  - Channel-surfing: changing the transmission frequency to a range where there is no interference from the attacker
  - Spatial retreat: moving to a new location where there is no interference

Jammers will be Punished

- A man skilled in the operation of commercial wireless Internet networks was sentenced for intentionally bringing down wireless Internet services across the region of Vernal, Utah.
- Ryan Fisher, 24, was sentenced to 24 months in prison to be followed by 36 months of supervised release, and to pay $65,000 in restitution.
- In total, more than 170 customers lost Internet service, some of them for as long as three weeks

Bottom Line



Don't be Evil !!

Sustaining Cooperation in Multi-Hop Wireless Networks

Motivation

- Free riding in multi-hop wireless networks
  - less contribution to the group
  - consume more than their fair share of a resource
- Routing protocols assume that nodes are well behaved
- Need for a system which discourages selfish behavior

Problem Definition

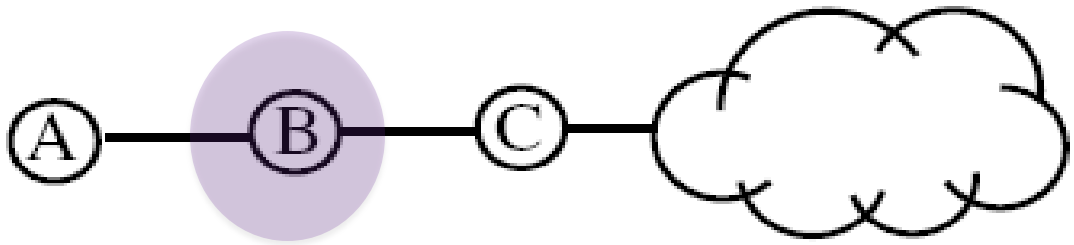
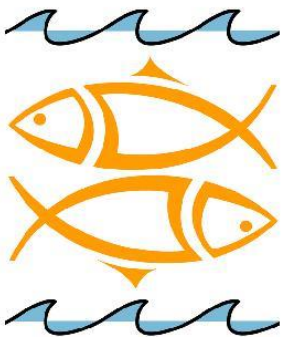


Figure 1: *An example multi-hop wireless network topology in which free-riding can take place.*

- B can avoid these forwarding loads in two distinct ways:
  - Forwarding level : drop packets for forwarding from *A*
  - Routing level : refuse to send routing messages that acknowledge connectivity with *A*

Assumptions

- Nodes → Selfish but not Malicious
- Most nodes in the system are cooperative
- Omni-directional radio transmitters and antennas
- Nodes have unforgeable ID

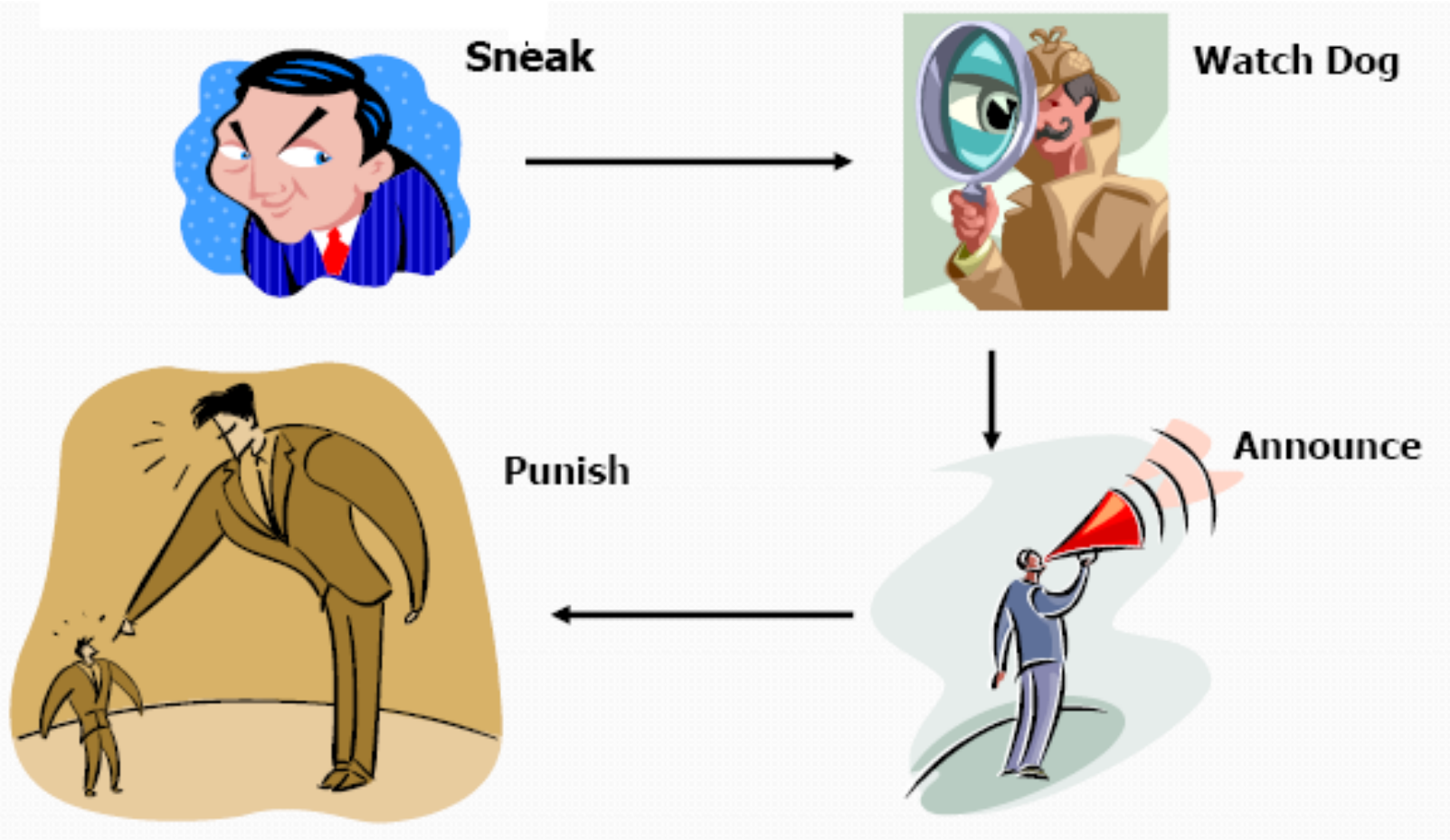


Catch !!

- Enforcement based mechanism to discourage free-riding through

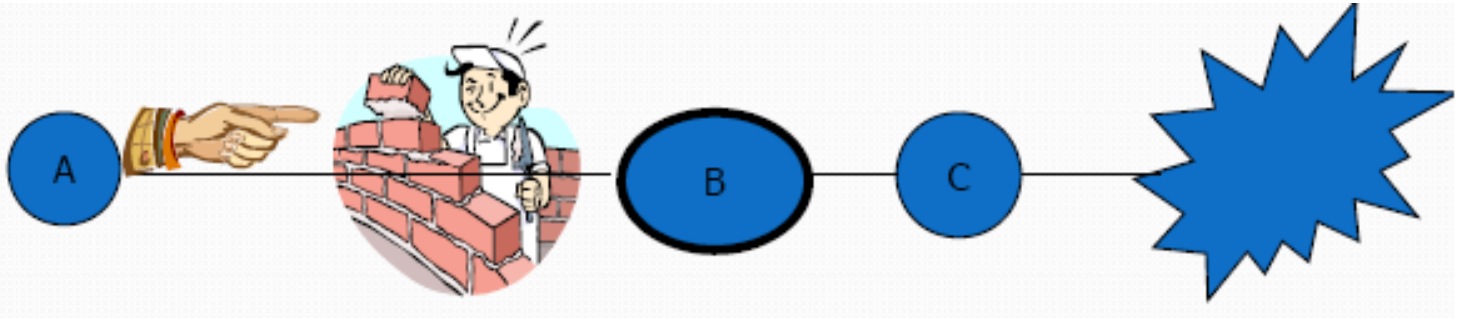
Fear of punishment !

Main Idea …



Problems in doing this??

- Distinguishing between Selfish nodes and Transmission errors
- Announcing the presence of free-rider to all, even if he/she is your only link to the outside world



Power of Anonymity

- The goal is to use cooperative nodes to monitor for the presence of free-riders and to isolate them
- Two problems
  - Distinguish them
  - Signal all of the free-rider's neighbor

Solution..

- Anonymous Challenge and Watchdogs
  - *To distinguish deliberate packet dropping from wireless errors*
- Anonymous Neighbor Verification
  - *To inform all other testers of the free-rider to isolate him*

Anonymous Challenges

- A watchdog is used
- All Testers regularly but unpredictably send an anonymous challenge to Testee for it to rebroadcast
  - A cryptographic hash of a randomly generated token
- Even a selfish testee must depend on at least one of its testers to forward its packets if it is to stay connected
- Any tester, without hearing the rebroadcast of its challenge → Flag Testee as a Free Rider, refuses to forward packets

Anonymous Neighbor Verification

- Once free-rider is detected, other testers must also be informed
- Challenge: the only path must be via the testee
- *Anonymous neighbor verification* (ANV) sub-protocol is defined

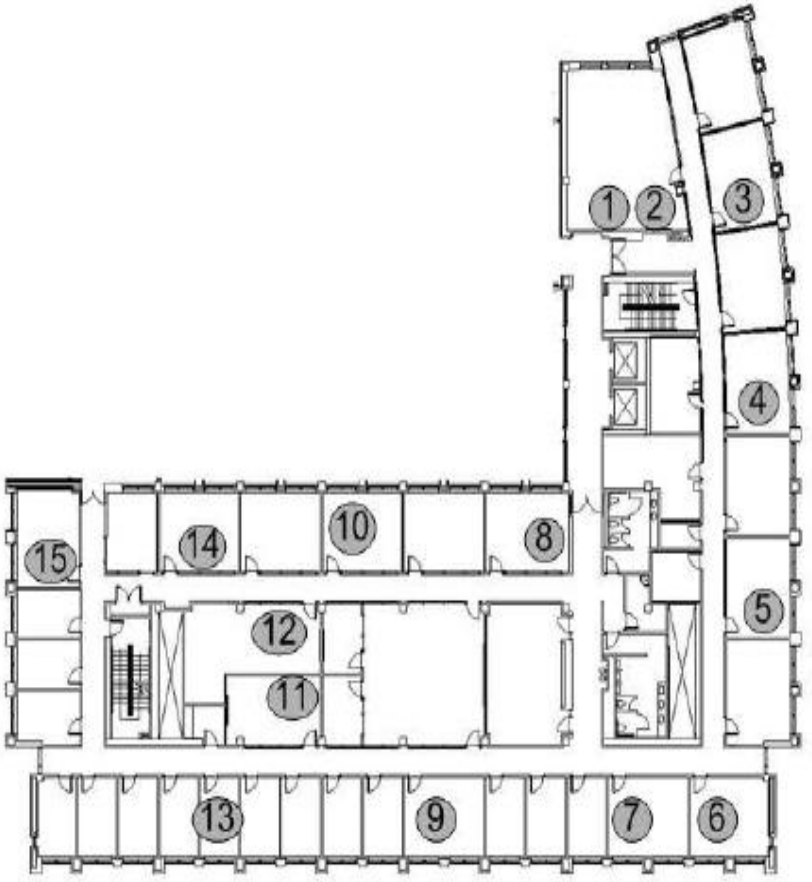
Anonymous Neighbor Verification

- First phase – ANV Open
  - Each tester sends cryptographic hash of a random token to the testee to rebroadcast
  - All others take note
- Second phase – ANV Close
  - If satisfied, release token to testee for it to rebroadcast
  - If senders don't receive tokens for all hash messages →

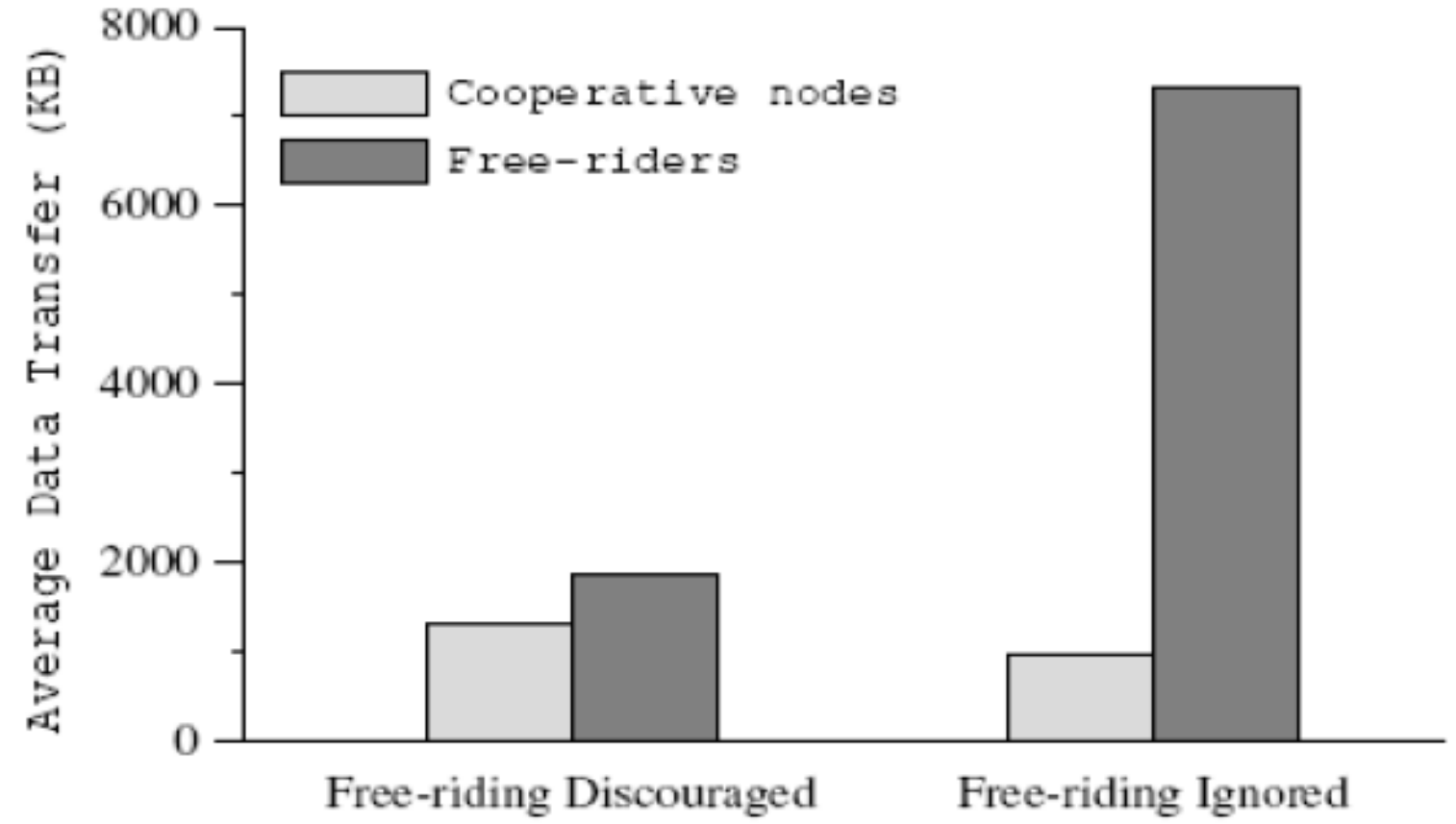
    infer that testee is free riding

Evaluation

- 15 PCs equipped with 802.11b
- Operating in the ad-hoc mode
- Diameter is between 3 and 5 hops
- Length of one epoch is set to one minute
- There are 15 anonymous *ACM* messages per epoch



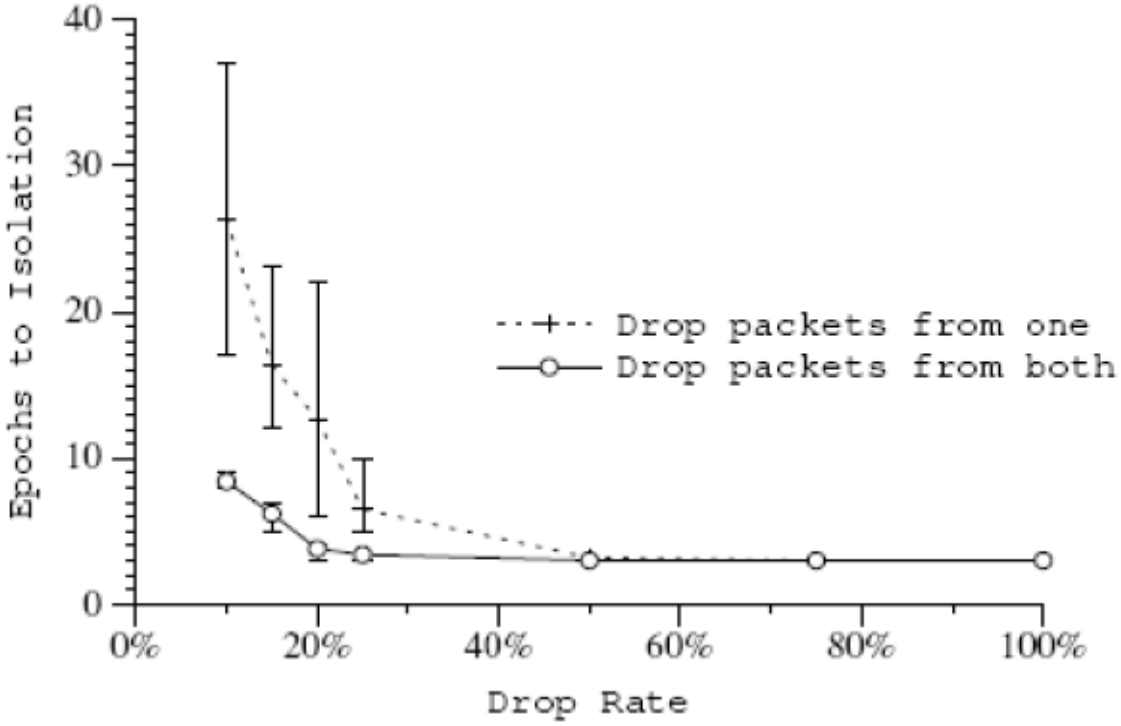
Evaluation


Average Data Transfer (KB)
8000
6000
4000
2000
0
Cooperative nodes
Free-riders
Free-riding Discouraged
Free-riding Ignored

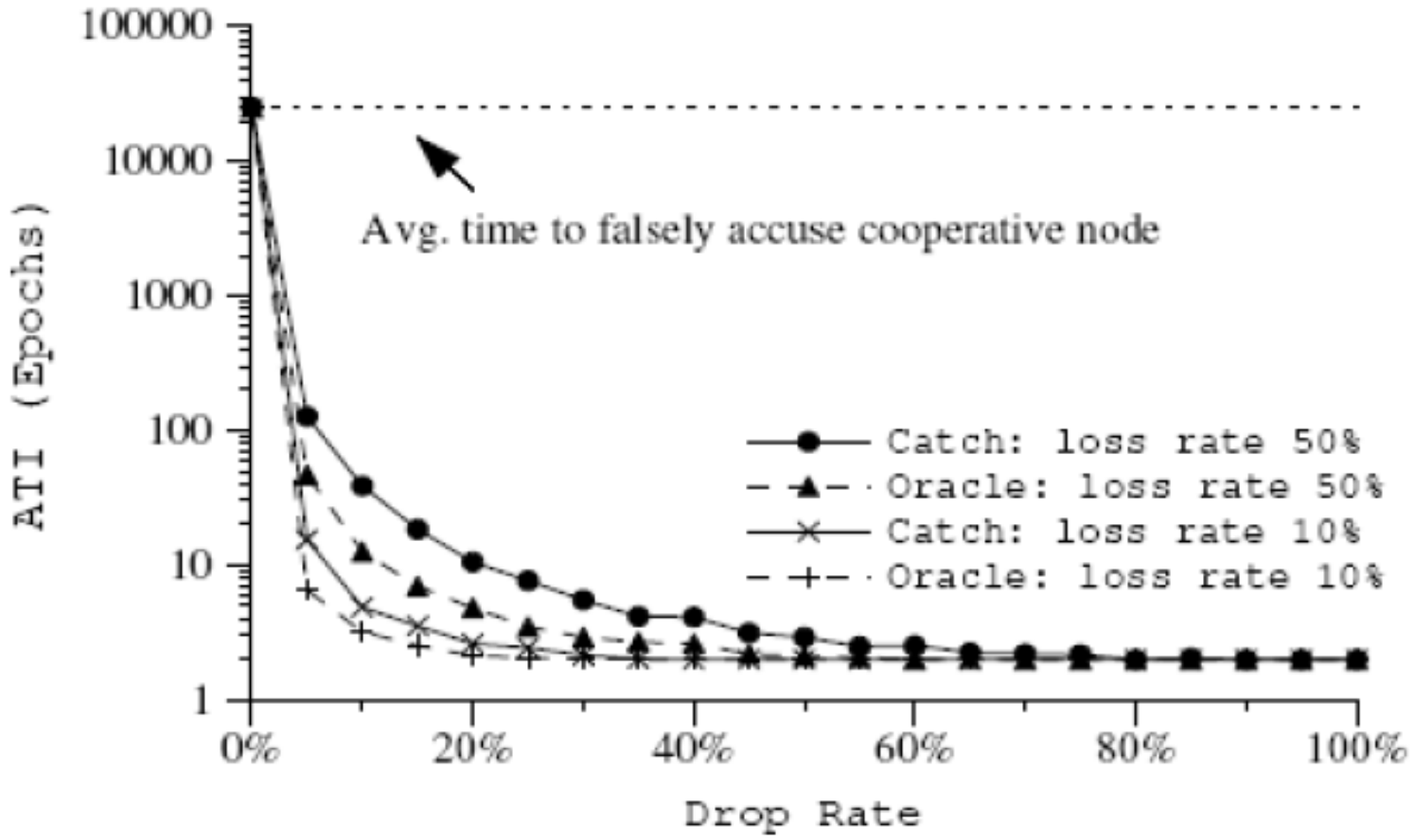
Evaluation

- Three nodes: the second one acts as free-rider
- The number of epochs required to detect free-riders in the testbed versus the fraction of packets a free-rider dropped

Each point is the average of 10 experiments



Evaluation



Average Time to Isolation (ATI)

Authors claim..

- No false positives (at least in the protocol)
- Low protocol overhead
- Works well under high network density
- Low impact of high wireless losses

Advantages…

- No confusion whether a node is free-riding or it just can't hear you
- Announcement by Silence !! (Passive)
- High risk in free-riding
- No way of selectively dropping packets due to anonymity
- Any Others??

Drawbacks

- Authors made their own simulator → Doubts reliability of their results..
- Others…

Conclusion

- Catch sustains cooperation with autonomous p nodes
- No central authority required unlike reputation/incentive based schemes
- No restrictions on workload, routing protocols or node objectives
- Isolates free-riders rather quickly
- No false positives

Outline

- General topics in network security
- Wireless Security
- IoT Security

Touch-And-Guard
Secure Pairing Through Hand Resonance

Wei Wang, Lin Yang, Qian Zhang

Hong Kong University of Science and Technology

Rise of Wearable/IoT Devices

- Rapid growth of wearable/IoT devices.
- Increasing chance of D2D communications in proximity.

Promising Applications & Security Threat

Mobile payment

Credit card, Transaction log, …

User-related, highly-sensitive data

Data sync

Biometrics, location info,…

First Step – Secure Pairing

Secret key: 00110…1010

Secret key: 00110…1010

Traditional Key - PIN Code

- No convenient input method for IoT/wearable device.
- Increasing security threats

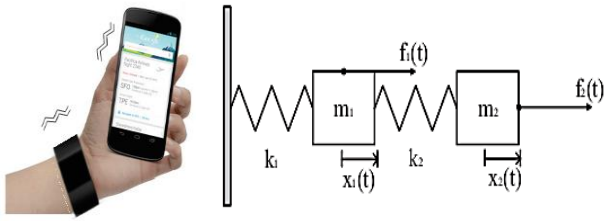
No convenient input

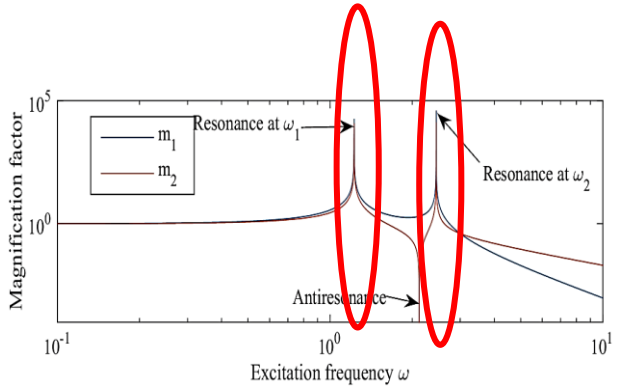
Shoulder surfing

Our solution - Touch-And-Guard (TAG)

- TAG generates shared secret bits from hand touch.
  - Commercial vibration motor and accelerometer
  - Simple & Intuitive
  - Secure to over-the-air eavesdroppers

Key Idea



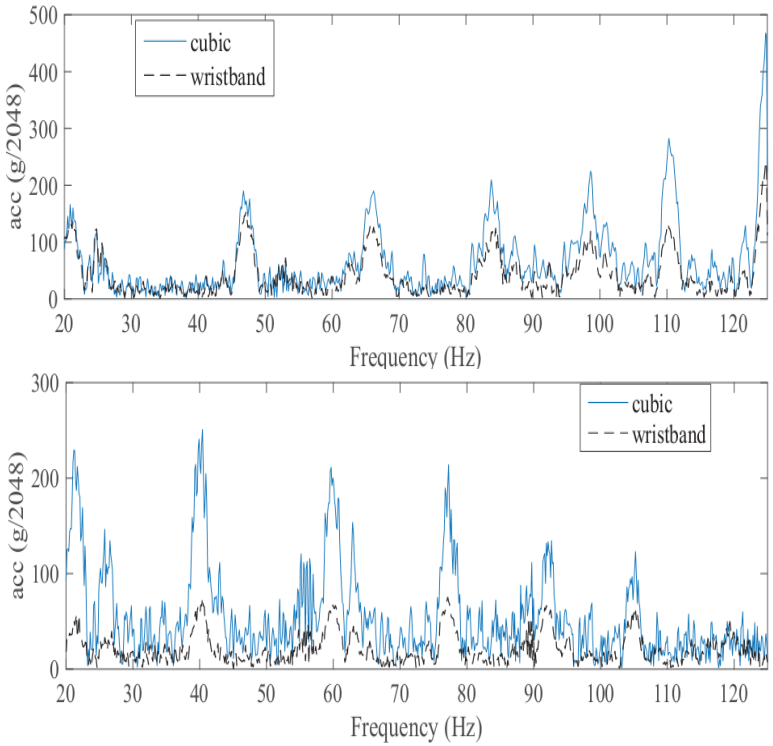


- Borrow from mechanical engineering
  - Hand touching a vibrating object forms a MDoF system:

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{f}$$

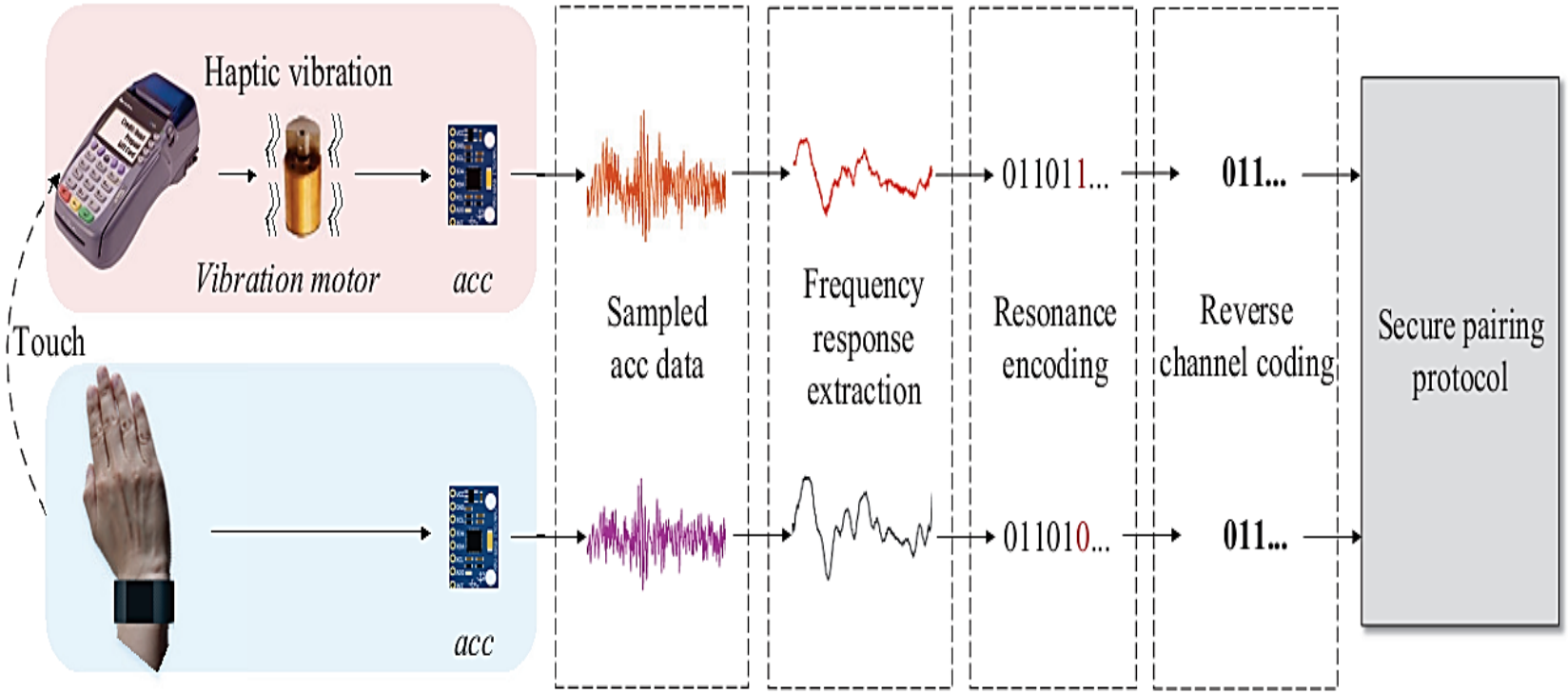
- Some observations
  - Their resonance frequency align perfectly.
  - More than one resonance frequency.

Resonance Frequencies as Secret Source

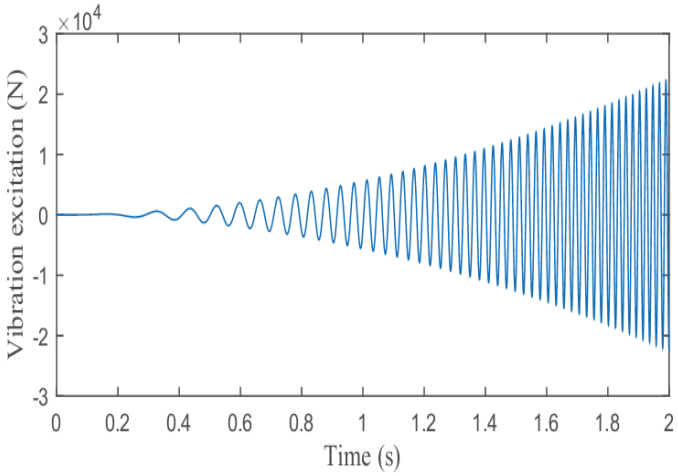


- Resonance frequencies (RF) are only obtainable to touched devices.
- One than one RF.
- The locations of RFs are random.
  - Posture
  - Strength
  - Touch position
  - …

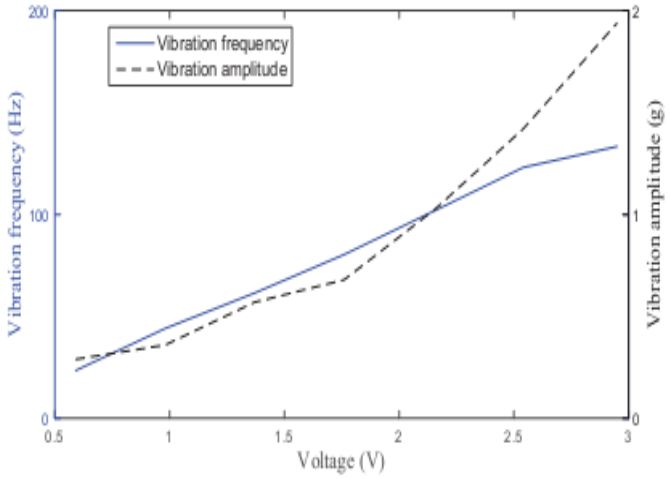
System Design



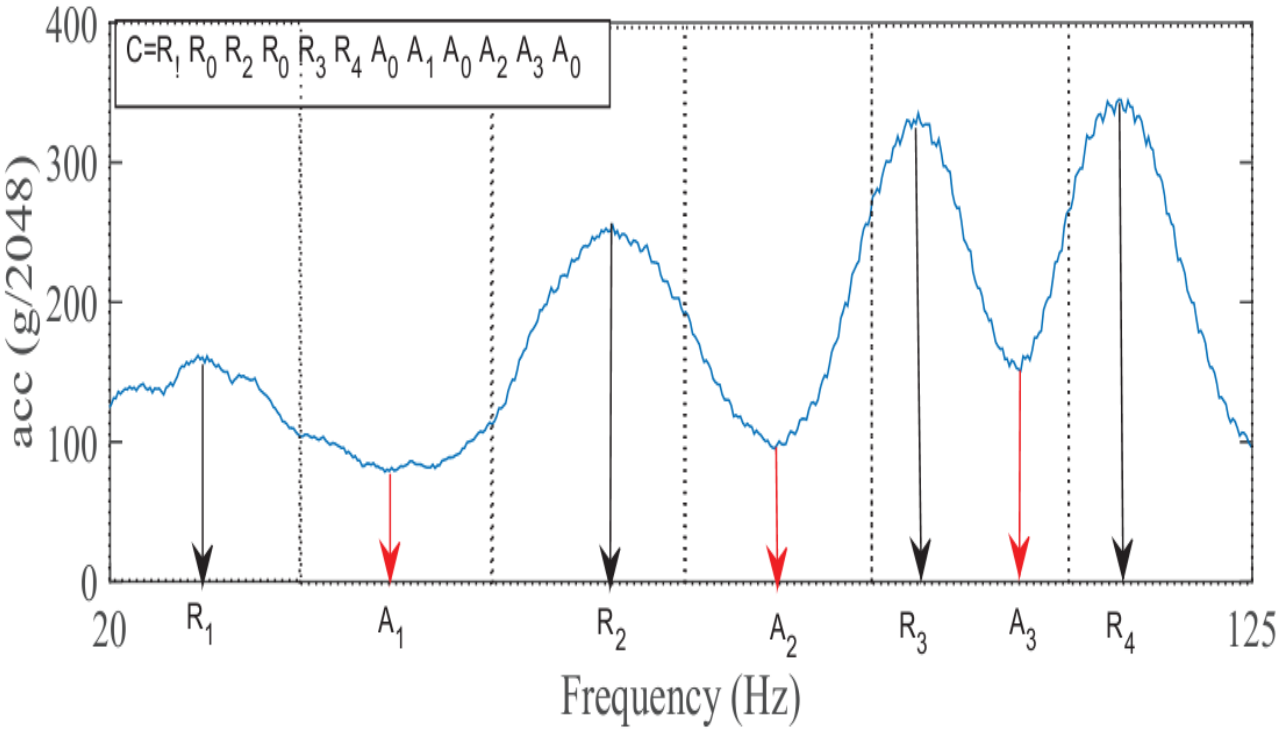
Vibration Excitation



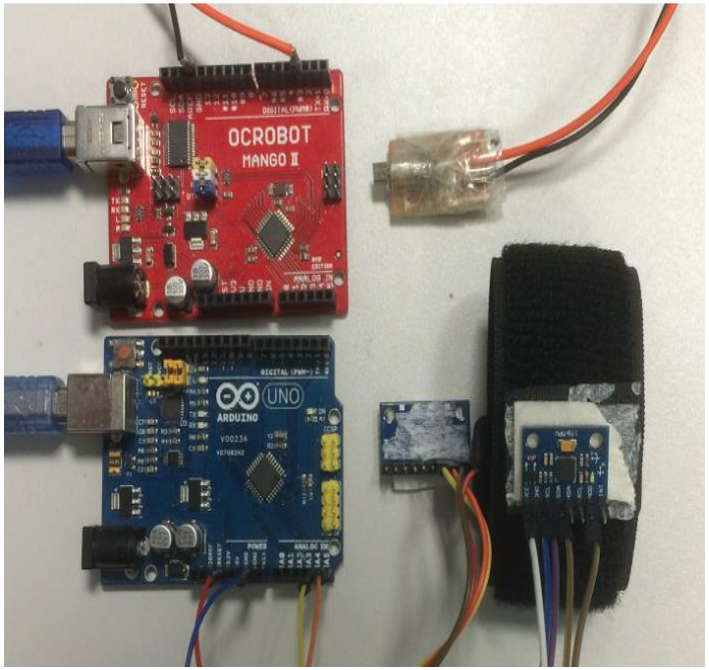
- Body RF ranges from several Hz to hundreds Hz
- 20~125 Hz, 1.75 second for our system.
- Frequency-sweeping excitation
- Eccentric-rotating mass vibrator controlled by PWM

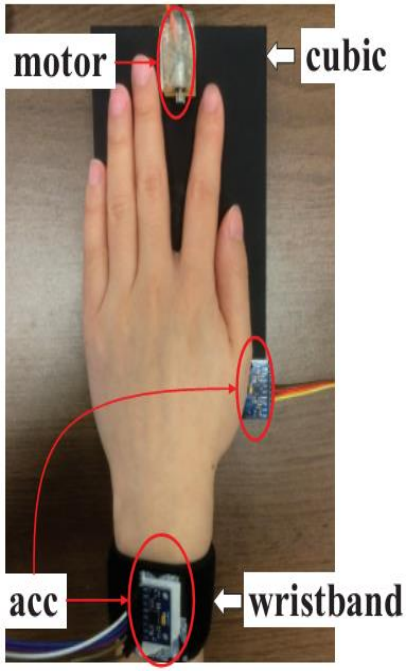


Resonance Encoding

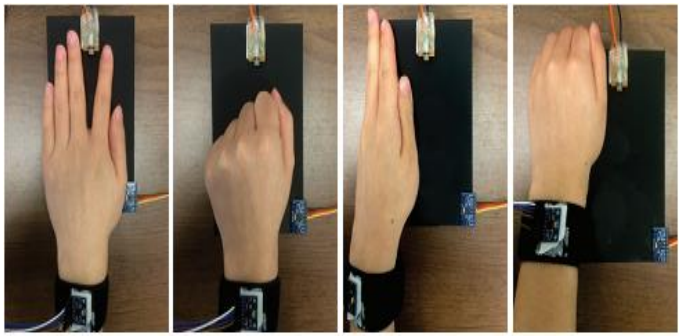


Prototype





Evaluation Setup

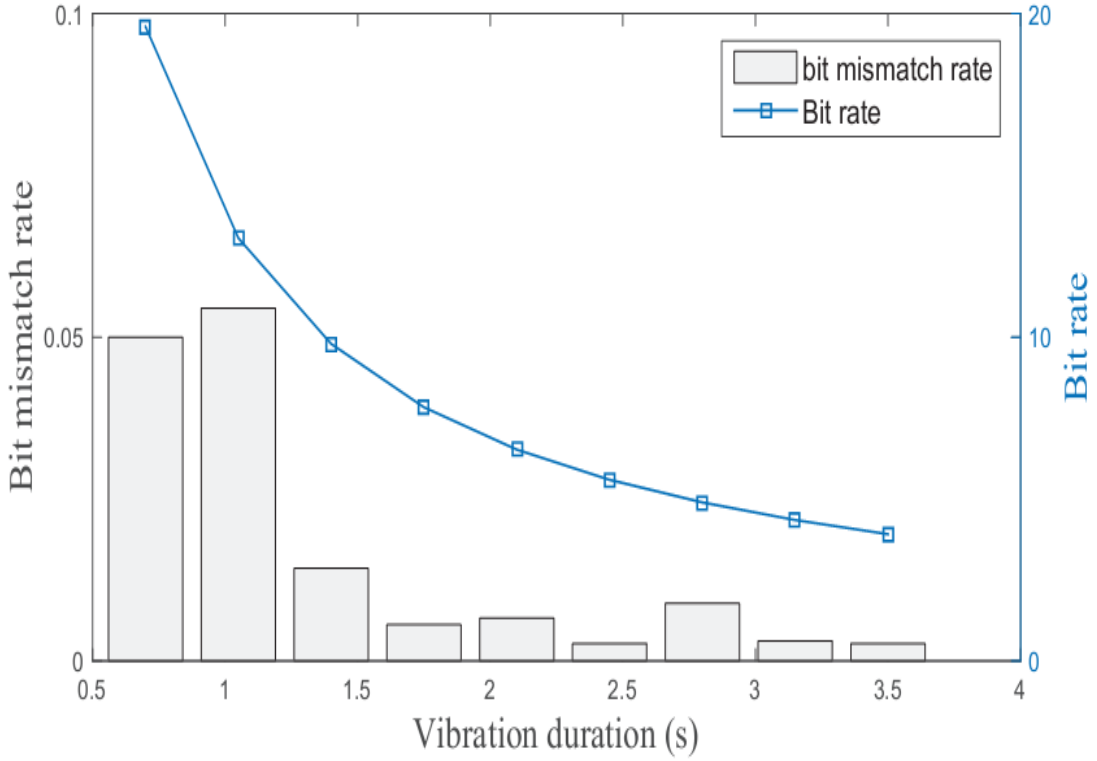


(a) Palm touch. (b) Fist touch. (c) Border touch. (d) Corner touch.

(a) Cubic box. (b) Smartphone. (c) Mouse. (d) Glass cup.

- 12 users (5 females + 7 males)
  - Age: 23 to 31.
  - BMI: 17.5 to 27.70
  - Wrist: 5.51 to 7.48 inches
- 4 Touch postures
- 30 trails for each posture.
- 1440 trials in total
- iPhone 5s to eavesdrop

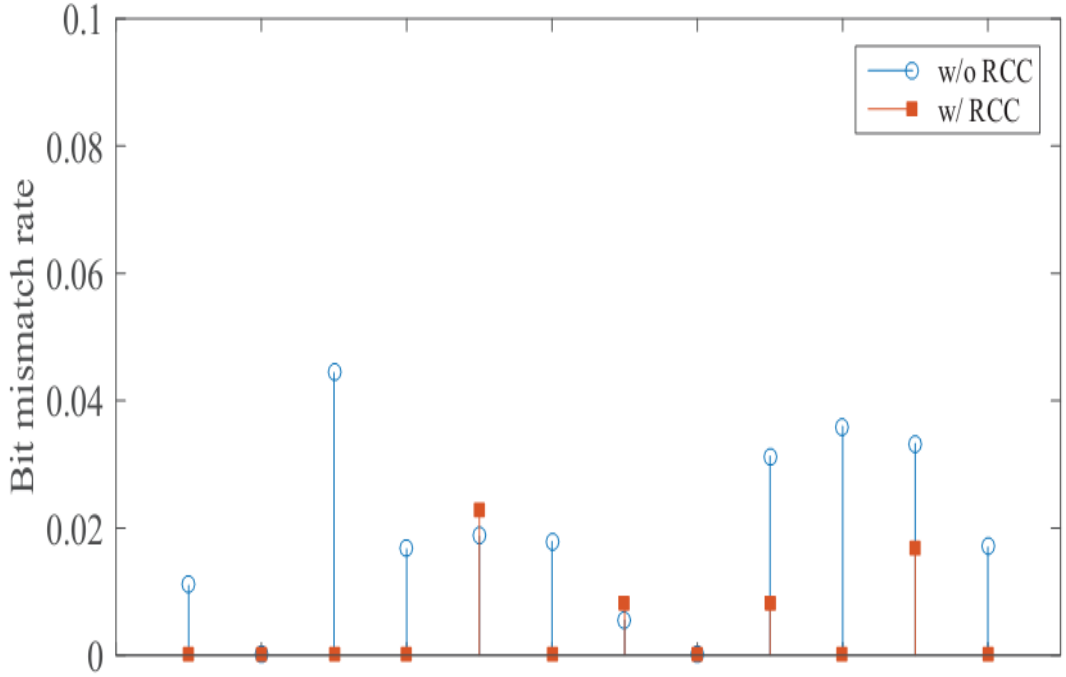
Pairing Performance vs. Vibration Duration



(b) w/ RCC.

- Longer vibration, less BER
- 1.75-second vibration is enough

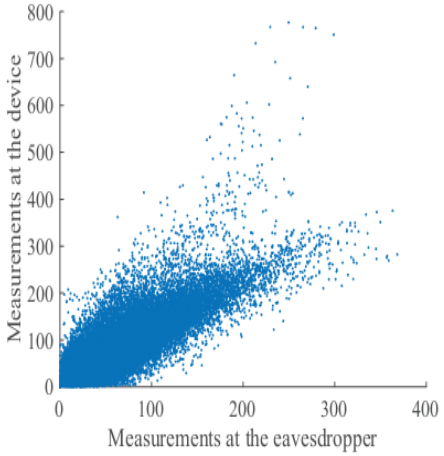
Pairing Performance of 12 Subjects



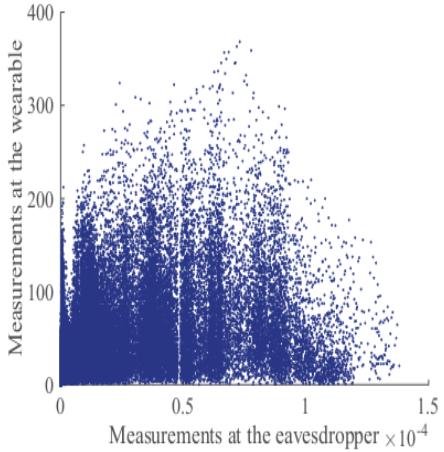
- The average bit mismatch rate is merely 0.5%
- The success rate of pairing is 96%

	Palm	Fist	Border	Corner
w/o RCC	1.13%	0.57%	2.1%	3.9%
w/ RCC	0	0	0.43%	1.47%

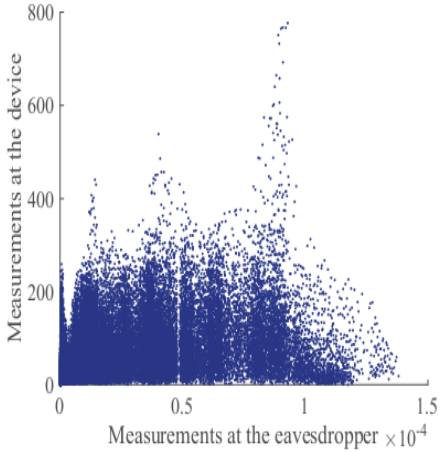
Resilience to Acoustic Eavesdropper



(a) Wearable vs. Device.



(b) Eavesdropper vs. Wearable.



(c) Eavesdropper vs. Device.

Eavesdropper can learn less than 0.01 bit from 1 secret bit.

Conclusion

- TAG is a system which securely pairing wearable devices.
- By exploiting the resonant frequency, TAG extracts secret bits from vibration.
- Extensive experiments validate our system
  - 58% faster than 4-bit PIN code
  - Robust in different scenarios

Friend or Foe? Your Wearable Devices Reveal Your Personal PIN

Lead Researcher: Yingying Chen

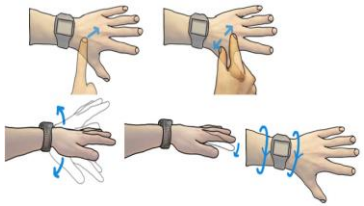
Chen Wang[†], Xiaonan Guo[†], Yan Wang*, Yingying Chen[†], Bo Liu[†]

[†]*Dept. of ECE, Stevens Institute of Technology*
** Dept. of CS, Binghamton University*

Motivation

- Wearable device

- Enable a broad range of useful applications
- Sensitive ir ıld ed



Electronic door lock

ATM machine

Keypad controlled server

Related Work

- Traditional attacks:

Shoulder surfing

Hidden camera

Keypad overlay

ATM Skimmer

- Audio-bas

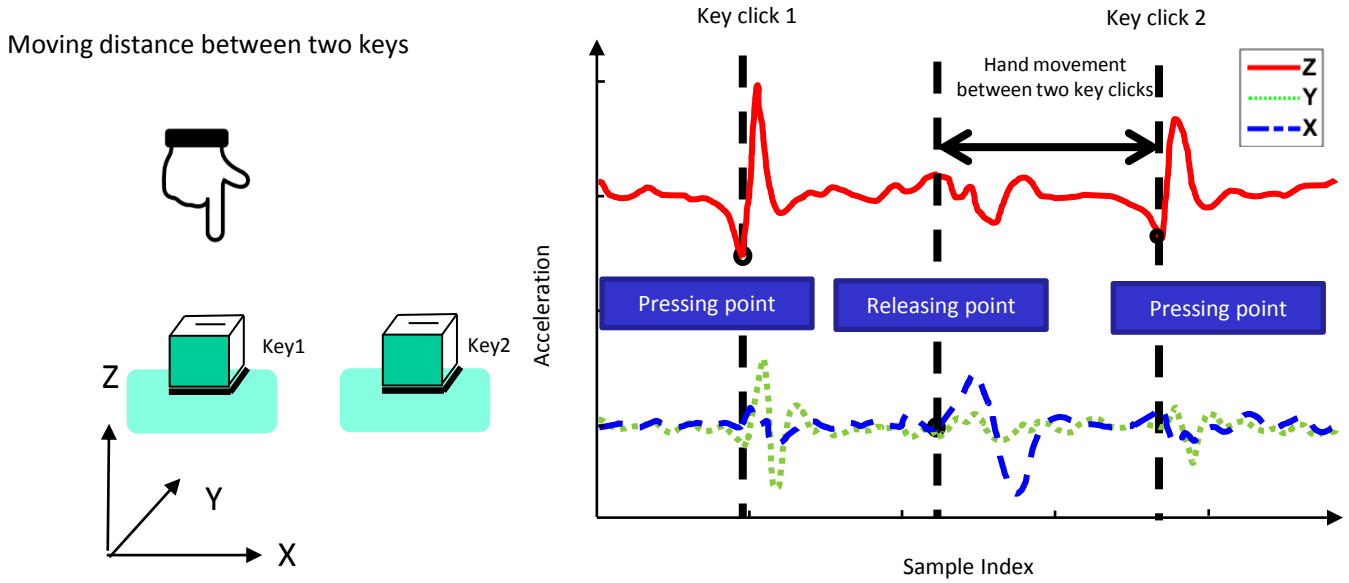
Require direct visual contact to key entry actions and additional installation efforts

- Concurrent smartwatch-based attacks
  - Hard to deal with non-contextual inputs such as PINs
  - Relies on a linguistic model and labelled training data Sensitive to environment noise
  -

Our goal: Training-free and contextual-free key entry inference system without additional devices and not subject to environmental noises.

Basic Idea

- Basic idea
  - Exploit embedded sensors in wearable devices
  - Capture dynamics of key entry activities
  - Derive fine-grained hand movement trajectories of key entries.



Attacking Scenarios

- Sniffing attacks
- Internal attacks

Device pairing using Bluetooth

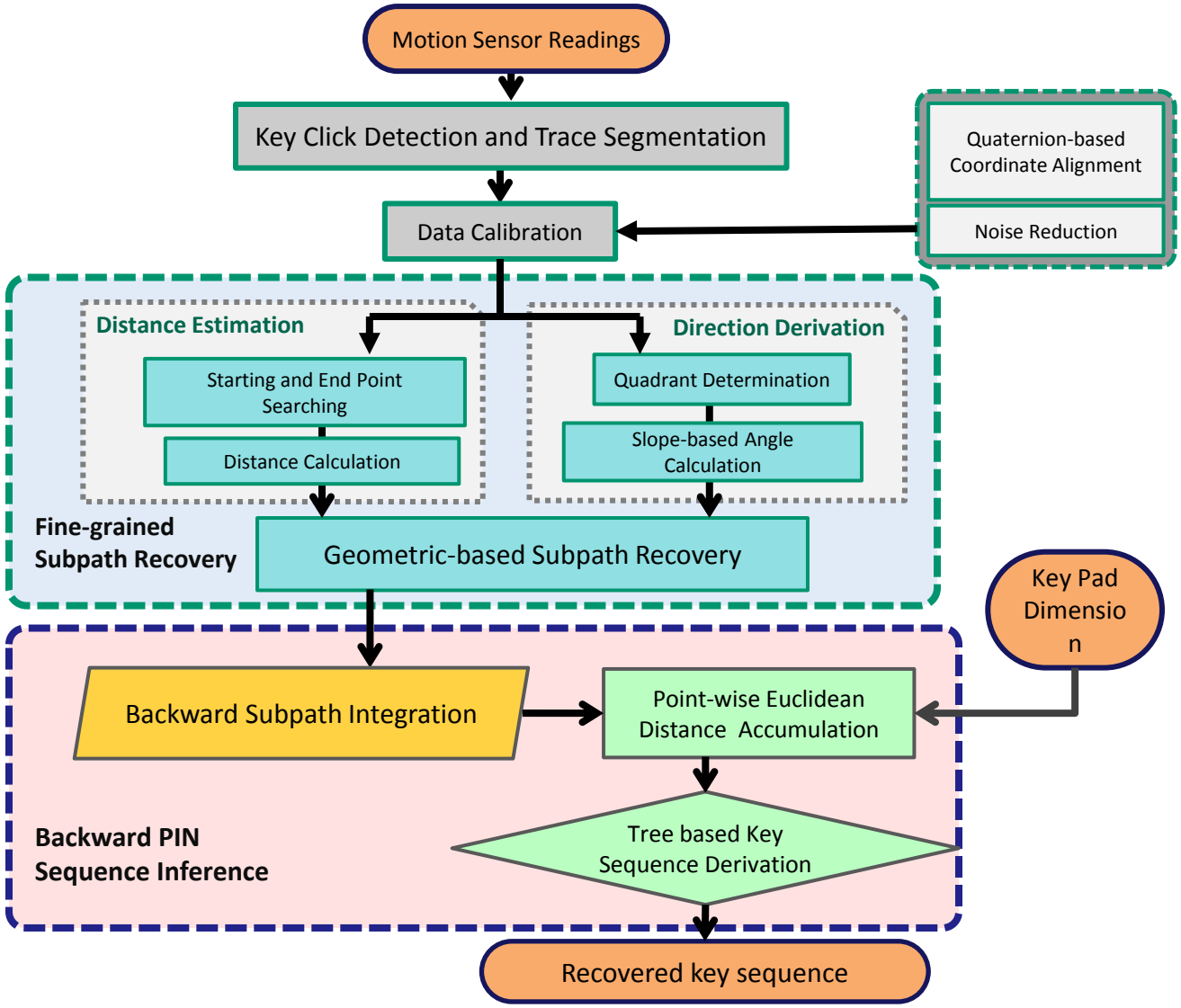
Bluetooth sniffing

Malwares

Challenges

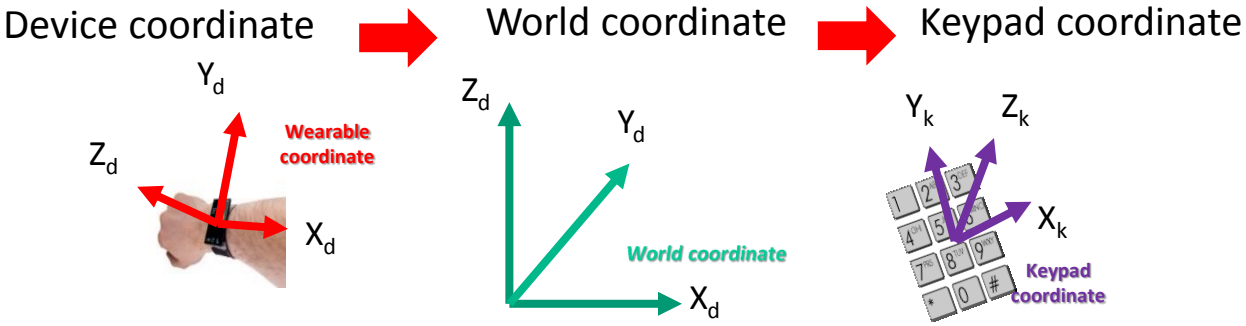
- Challenges
  - Robust fine-grained hand movement tracking
  - Training free key entry recognition
  - Recovering PIN sequence without contextual information
  - Sensing with single free-axis wearable device

Framework Overview



Quaternion-based Coordinate Alignment

- Quaternion



$$\vec{a}_w = q_{dw}\vec{a}_d q_{dw}^{-1}$$

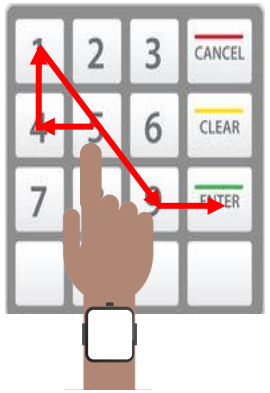
$\vec{a}_w$ Sensor reading in world coordinate

$\vec{a}_d$ Sensor reading in device coordinate

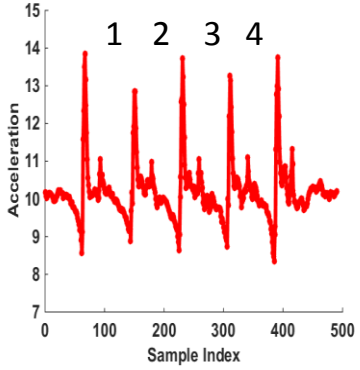
Quaternion q_{dw} **conversion from the world coordinate to keypad coordinate**

Fine-grained Subpath Recovery

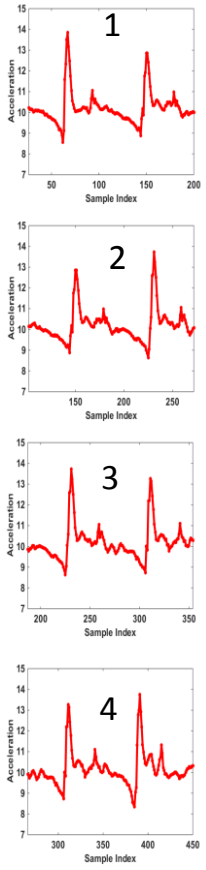
Input "5419-Enter"



Subpaths



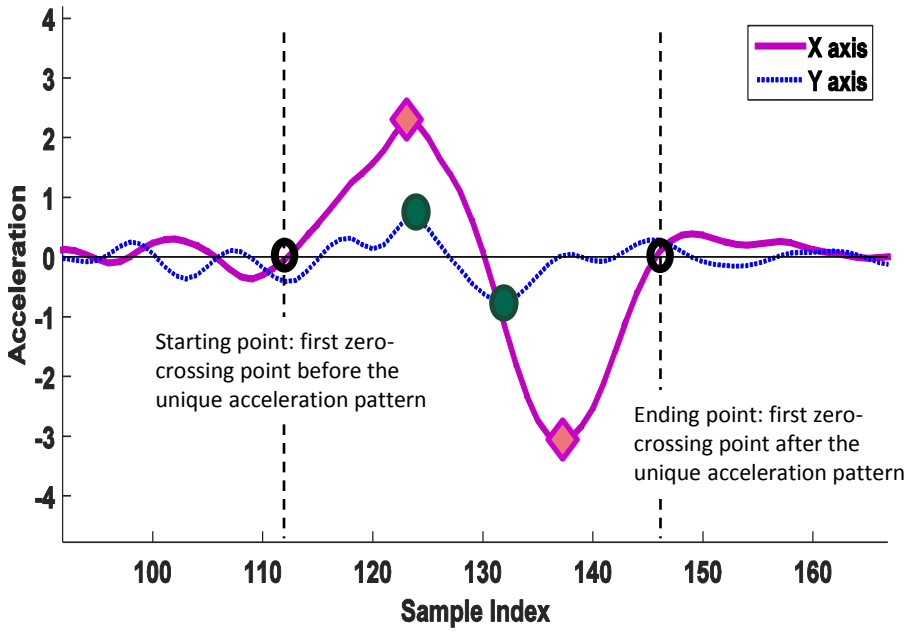
Key-click trace segmentation



Subpath recovery

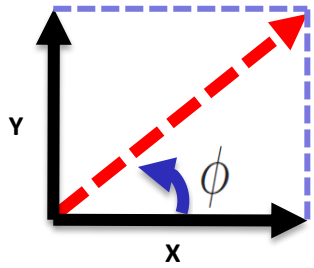
Subpath Distance Estimation

- Starting and ending points searching based on pressing and releasing points



- Distance ca
  - Double

Subpath Direction Derivation

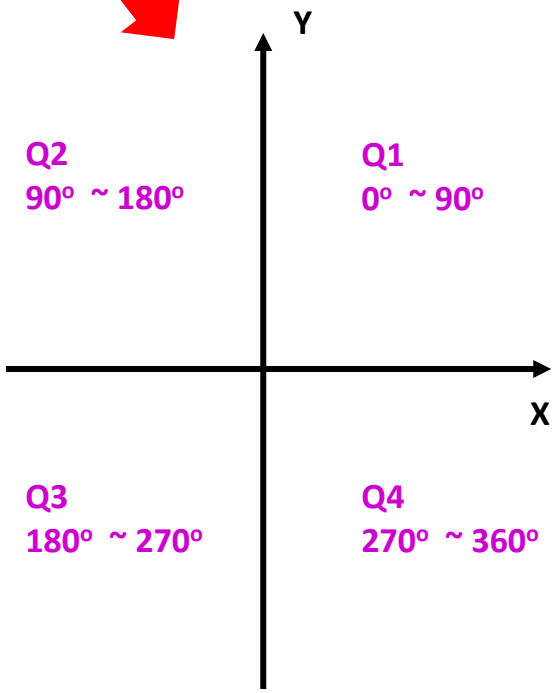


$$\phi = \left| arctan\left(\frac{s_y}{s_x}\right) \right|$$

Range 0° ~ 90°

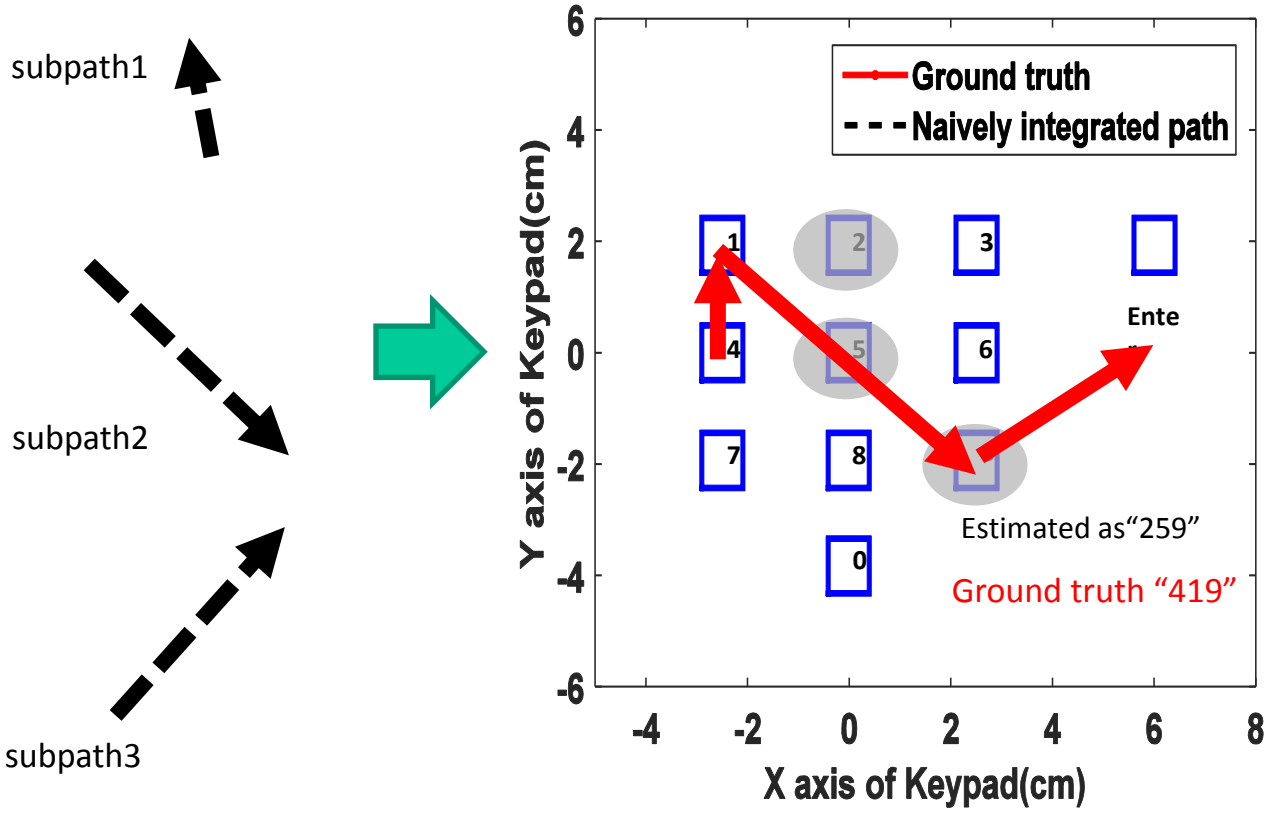
Quadrant Determination

$$\vartheta = \begin{cases} \phi; & if\ Q = 1, \\ 180° - \phi; & if\ Q = 2, \\ 180° + \phi; & if\ Q = 3, \\ 360° - \phi; & if\ Q = 4. \end{cases}$$



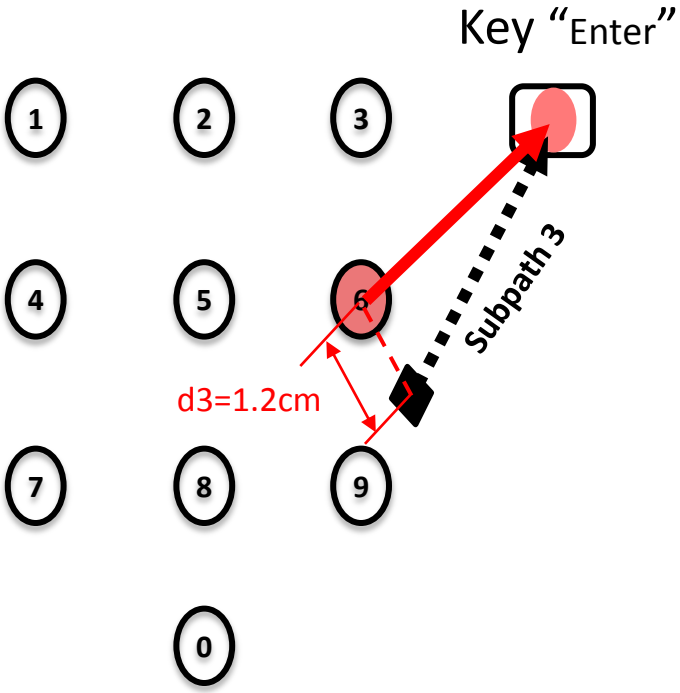
Backward PIN Sequence Inference

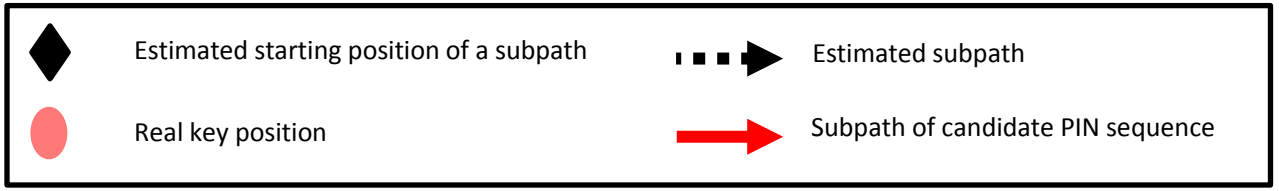
- Backward Subpath Integration



Point-wise Euclidean Distance Accumulation

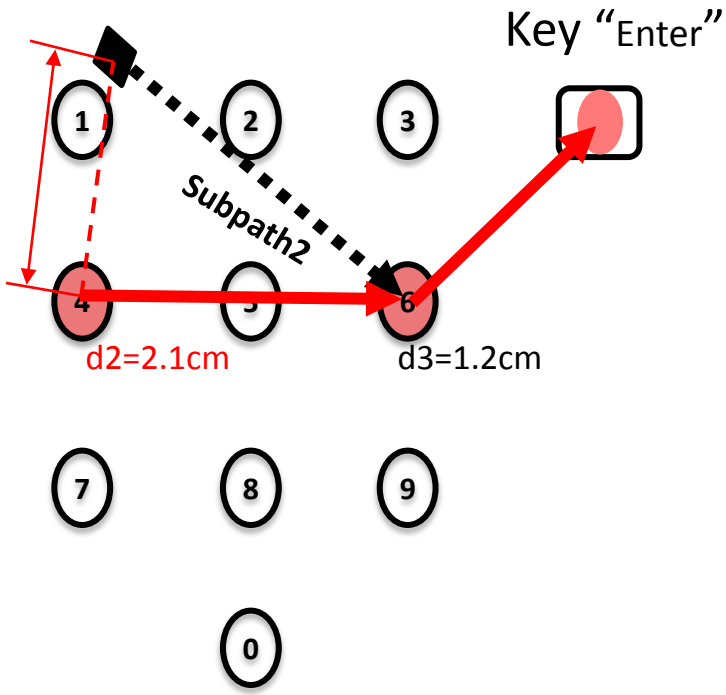
- Example of candidate sequence 846
  - The third subpath:
    - d3= 1.2cm
    - D3=d3=1.2cm





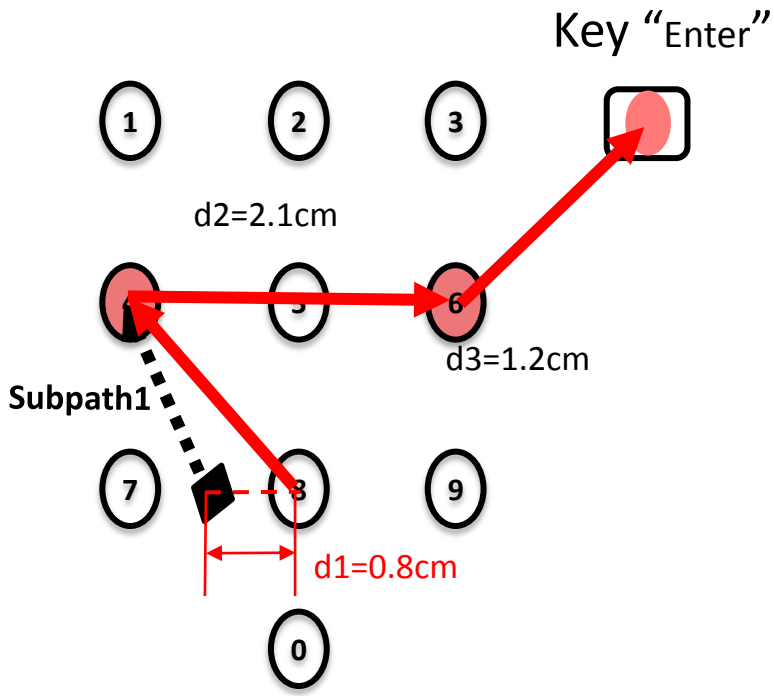
Point-wise Euclidean Distance Accumulation

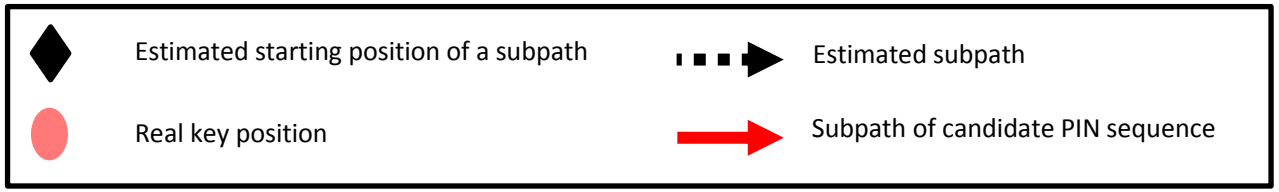
- Example of candidate sequence 846
  - The third subpath:
    - d3= 1.2cm
    - D3=d3=1.2cm
  - The second subpath:
    - d2=2.1cm
    - D2=D3+d2=3.3cm



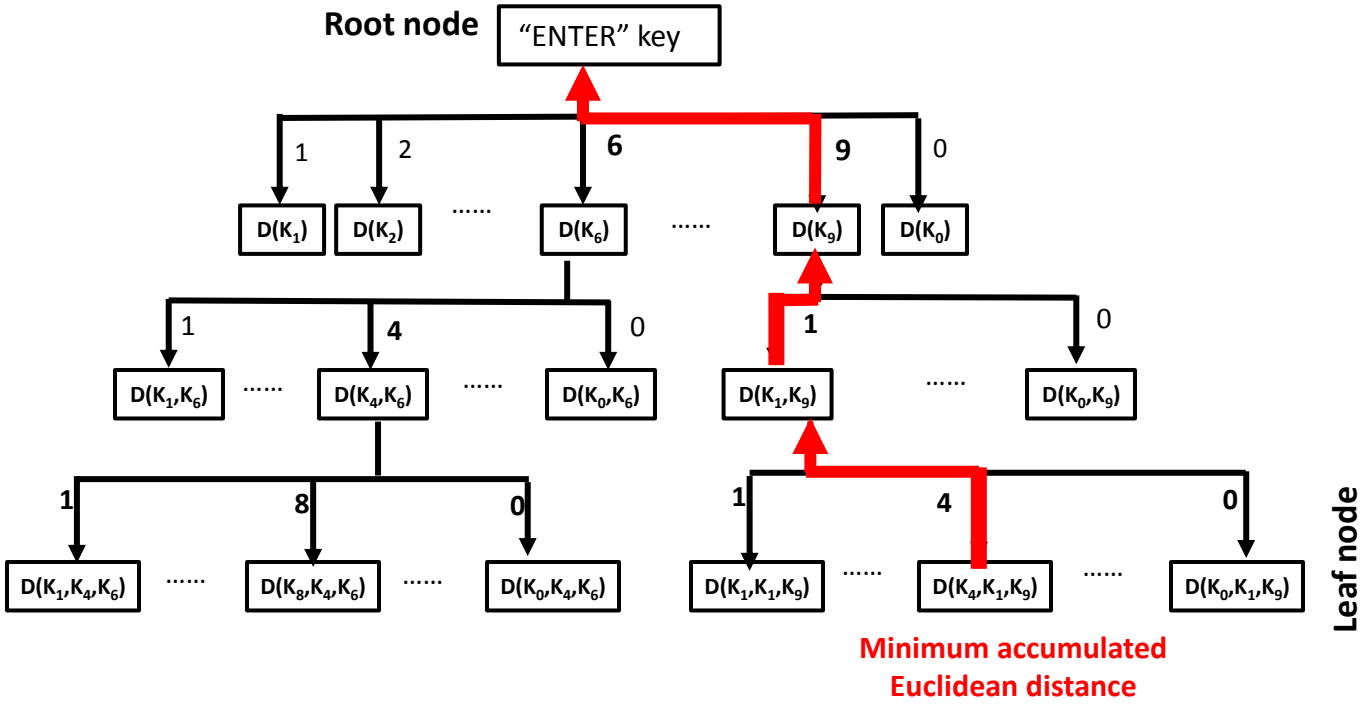
Point-wise Euclidean Distance Accumulation

- Example of candidate sequence 846
  - The third subpath:
    - d3= 1.2cm
    - D3=d3=1.2cm
  - The second subpath:
    - d2=2.1cm
    - D2=D3+d2=3.3cm
  - The first subpath
    - d1=0.8cm
    - **D1=D2+d1=4.1cm**





Tree-based Key Sequence Inference



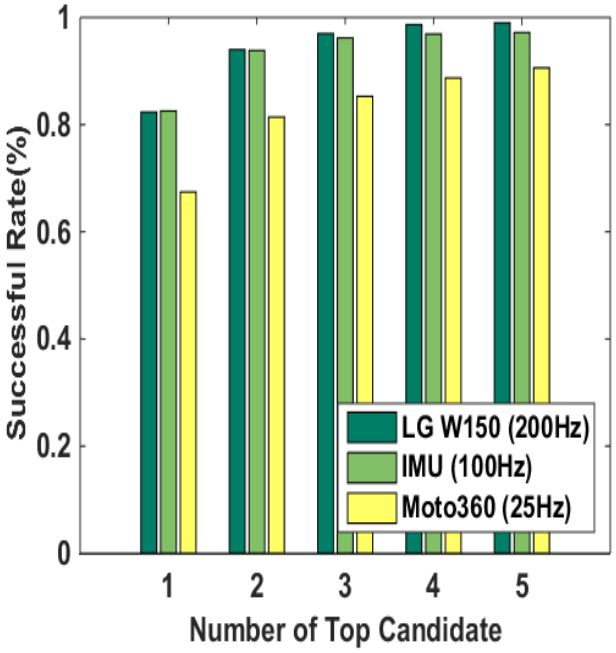
Revealed PIN sequence: "419"

Experimental Methodology

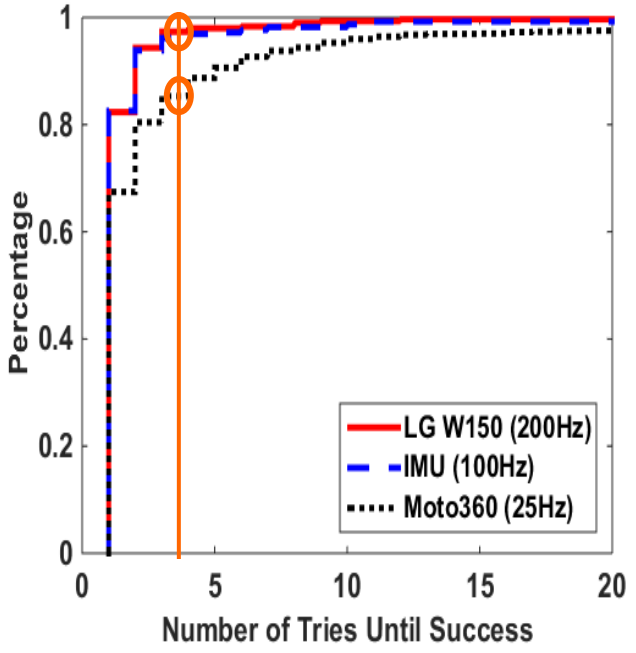
- Three Keypads
  - Real ATM machine
  - Detachable ATM pad
  - Keyboard number pad
- Three wearable Devices
  - LG150 (200Hz)
  - Moto360 (25Hz)
  - Invensense MPU-9150 (100Hz)
- Data collection
  - Number of volunteers: 20
  - Key-entry: 4-digit PIN sequences (5 key clicks)
- Evaluation Metrics: Top-k success rate, number o

Performance of Different Wearable Devices

- Performance of Backward PIN-Sequence Inference with three kinds of wearables on the detachable ATM Keypad



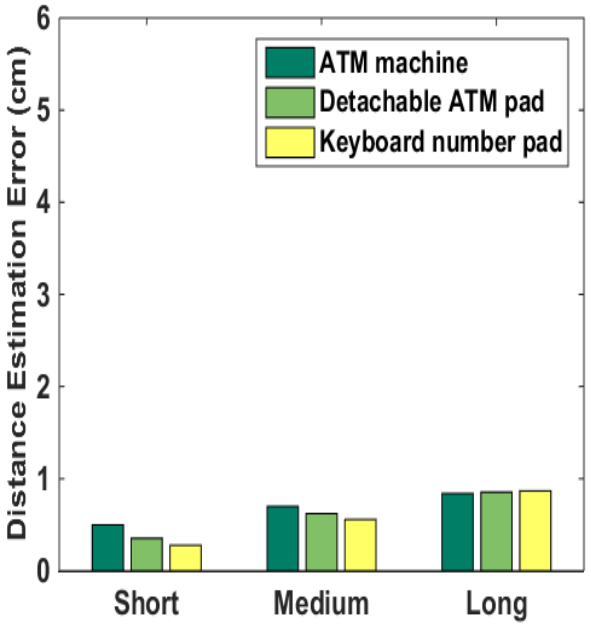
Higher sampling rate leads to higher successful rate



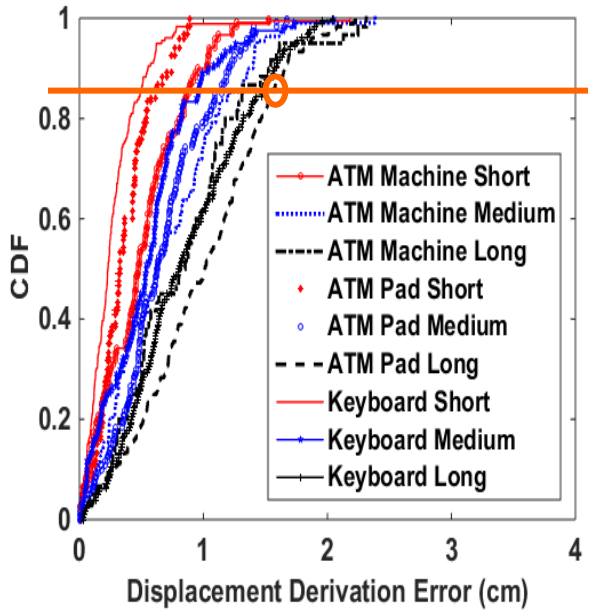
Adversary can break over 97% PIN entries from the LG W150 and IMU within 5tries. 90% for Moto 360.

Distance Estimation

- Fix 100 Hz sampling rate, testing 2.5cm (Short), 5cm (Medium) and 6.4cm (Long) moving distance



The mean error is only in mm-level



80th percentile errors are less than 1.5cm

Conclusion

- Wrist-worn wearable devices can be exploited to recover user's fine-grained hand movements during key-entry activities
- Present a PIN-sequence inference framework to recover the user's secret key entries without requiring any training or contextual information
- The system devises a Backward PIN-sequence Inference Algorithm to reveal user's secret PINs
- Extensive experiments show that our system can achieve high accuracy in revealing the user's PIN sequences with one or within three tries