CIS373 - Pervasive Computing Security/Privacy in IoT

Erik Fredericks // frederer@gvsu.edu

Adapted from materials provided by Xiang Cao

Why be concerned about IoT?

It’s more than just another computer, right?

- All of the same issues we have with access control, vulnerability management, patching, monitoring, etc.
- Imagine your network with 1,000,000 more devices
- Any compromised device is a risk on the network

Let's start here!

Massive DDoS Attack on Dyn DNS

https://www.youtube.com/watch?v=GUeeR4B6V_A

Or how about session-jacking?
https://www.youtube.com/watch?v=O3NAM8oG1WM

A bit old, but imagine something passively listening for session IDs....

- How do you fix this one?

Internet of Things Research Study

By Craig Smith and Daniel Miessler, HP Security Research

10 most popular IoT devices in different categories:

TV, webcam, home thermostat, remote power outlet, sprinkler controller, hub for controlling multiple devices, door lock, home alarm, scales, garage door opener

Internet of Things Research Study: Authentication

8 failed to require passwords of sufficient complexity or length.

- Most allowed e.g., “1234” or “123456”

Internet of Things Research Study: Privacy

9 collected at least one piece of personal information via the device, its cloud, or the app

- E.g., name, address, date of birth, health data, even credit card numbers

Internet of Things Research Study: Encryption

7 had **unencrypted communications** with Internet or local network.

- Half of mobile apps had unencrypted communications.

Internet of Things Research Study: web user interface

6 had user interface security problems

- e.g., poor session management, weak default credentials, credentials transferred in clear

Internet of Things Research Study: Software updates

6 didn’t use encryption to upload software updates.

- Some updates could be intercepted and the whole code viewed and changed.

Smartwatches

10 of the top smartwatches in today’s market
- Android or iOS mobile device and app

9 of 10: watch communications trivially intercepted
7 of 10: firmware transmitted without encryption

How Safe are Home Security Systems?

10 existing home security systems
- 7 with cloud interface, all with mobile interface

10 of 10 vulnerable to brute-force password-guessing attack

OWASP recommendations: Privacy

- Only collect data the device needs to function
- Try not to collect sensitive data
- De-identify or anonymize
- Ensure the Thing and its components protect personal information
- Only give access to authorized individuals
- “Notice and Choice” for end-users if more data is collected than would be expected

Open Web Application Security Project
https://www.owasp.org/index.php/OWASP_Internet_of_Things_Top_Ten_Project

OWASP recommendations: Authentication

- Require strong passwords
- Protect credentials
- 2-factor authentication where practical
- Secure password recovery mechanisms
- Re-authentication for sensitive features
- Password control configuration options

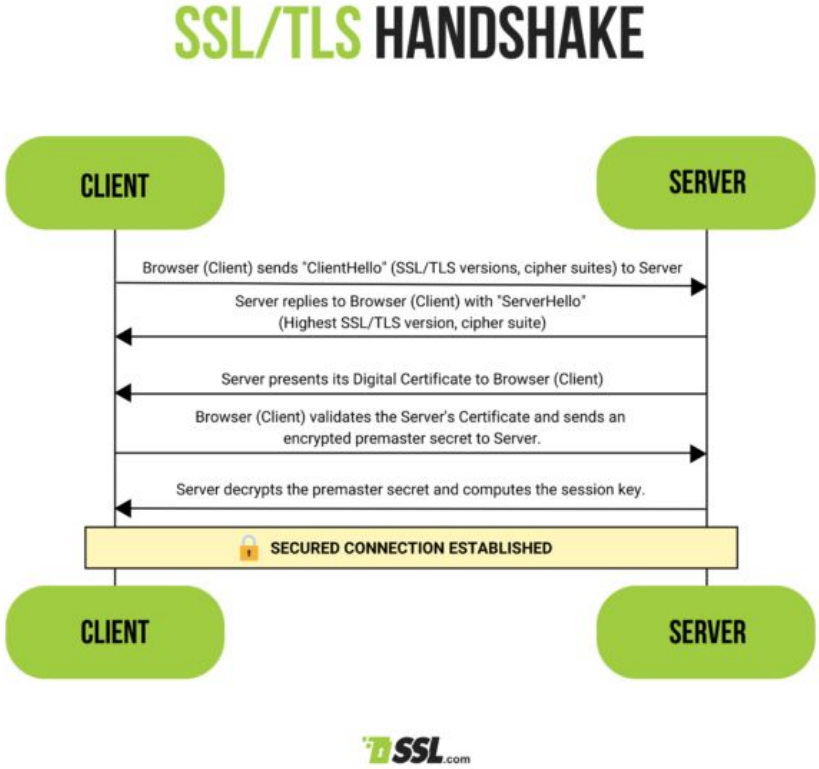
Open Web Application Security Project
https://www.owasp.org/index.php/OWASP_Internet_of_Things_Top_Ten_Project

OWASP recommendations: Transport encryption

Encrypt data when transiting networks

Use SSL/TLS (Secure Sockets Layer/Transport Layer Security), or other industry standards if these are not available

Open Web Application Security Project
https://www.owasp.org/index.php/OWASP_Internet_of_Things_Top_Ten_Project



OWASP recommendations: Web user interface

Change default passwords during initial setup – ideally also default usernames
Robust password recovery mechanisms
Don’t expose credentials in network traffic
Require strong passwords
Lockout account after 3-5 failed logins

Open Web Application Security Project
https://www.owasp.org/index.php/OWASP_Internet_of_Things_Top_Ten_Project

OWASP recommendations: Software/firmware updates

Ensure updates are possible!
Encrypt the update file
Transfer update over encrypted connection
Ensure update file doesn’t expose sensitive info
Verify update before uploading and applying
Secure the update server

Open Web Application Security Project
https://www.owasp.org/index.php/OWASP_Internet_of_Things_Top_Ten_Project

OWASP IoT Top 10

Category	IoT Security Consideration	Recommendations
I1: Insecure Web Interface	•Ensure that any web interface coding is written to prevent the use of weak passwords …	When building a web interface consider implementing lessons learned from web application security. Employ a framework that utilizes security …
I2: Insufficient Authentication/Authorization	•Ensure that applications are written to require strong passwords where authentication is needed …	Refer to the OWASP Authentication Cheat Sheet
I3: Insecure Network Services	•Ensure applications that use network services don't respond poorly to buffer overflow, fuzzing …	Try to utilize tested, proven, networking stacks and interfaces that handle exceptions gracefully...
I4: Lack of Transport Encryption	•Ensure all applications are written to make use of encrypted communication between devices…	Utilize encrypted protocols wherever possible to protect all data in transit…
I5: Privacy Concerns	•Ensure only the minimal amount of personal information is collected from consumers …	Data can present unintended privacy concerns when aggregated…
I6: Insecure Cloud Interface	•Ensure all cloud interfaces are reviewed for security vulnerabilities (e.g. API interfaces and cloud-based web interfaces) …	Cloud security presents unique security considerations, as well as countermeasures. Be sure to consult your cloud provider about options for security mechanisms…
I7: Insecure Mobile Interface	•Ensure that any mobile application coding is written to disallows weak passwords …	Mobile interfaces to IoT ecosystems require targeted security. Consult the OWASP Mobile …
I8: Insufficient Security Configurability	•Ensure applications are written to include password security options (e.g. Enabling 20 character passwords or enabling two-factor authentication)…	Security can be a value proposition. Design should take into consideration a sliding scale of security requirements…
I9: Insecure Software/Firmware	•Ensure all applications are written to include update capability and can be updated quickly …	Many IoT deployments are either brownfield and/or have an extremely long deployment cycle...
I10: Poor Physical Security	•Ensure applications are written to utilize a minimal number of physical external ports (e.g. USB ports) on the device…	Plan on having IoT edge devices fall into malicious hands...

Another example of a hardware attack!

Stuxnet: https://vimeo.com/25118844

And:
https://www.youtube.com/watch?v=Fe581bHpvZo

Attack Taxonomy

- Attacks based on Information Disruption
- Attacks based on Host properties
- Attacks based on Network properties

Attacks based on Information Disruption

Interruption
- Denial-of-service attack.
- Communication links lost or made unavailable

Interception
- Eavesdrop on the information to threaten data privacy and confidentiality

Modification
- Tamper medical information

Fabrication
- Forge or inject false information

Attacks based on Host properties

User Compromise (social engineering)

- Compromise a user's health device or network
- Mostly involves revealing passwords, cryptographic keys or user data
- https://www.youtube.com/watch?v=lc7scxvKQOo

Hardware Compromise

- Physically tamper the device
- Extract on – device program code, keys and data
- Reprogram with false program

Software Compromise

- Forces malfunction by taking advantages of the vulnerabilities in either the operating system or other applications of the device

Attacks based on Network properties

Standard Protocol Compromise

- An attacker deviates from standard protocols
- Acts maliciously to threaten service availability, message privacy, integrity, and authenticity

Network Protocol Stack Attack

- Attack on protocol itself (denial of service, man in the middle, etc.)

Threat vs. Opportunity

If misunderstood and misconfigured, IoT poses risk to our data, privacy, and safety

If understood and secured, IoT will enhance communications, lifestyle, and delivery of services

Case study: WSN security

Privacy

Sensor networks raise new threats that are qualitatively different from what private citizens worldwide faced before

- Sensor networks allow data collection, coordinated analysis, and automated event correlation

- Networked systems of sensors can enable routine tracking of people and vehicles over long periods of time
  - Think: smart cities, VANETs, personal monitoring devices

Problems Applying Traditional Network Security Techniques

Sensor devices are limited in their energy, computation, and communication capabilities

Sensor nodes are often deployed in open areas, thus allowing physical attack

Sensor networks closely interact with their physical environments and with people, posing new security problems

Key Establishment Solutions

Sensor devices have limited computational power, making public-key cryptographic primitives too expensive in terms of system overhead.

(1) Simplest solution is a network-wide shared key

- All sensor nodes use the same key

Discussion: What is the problem of this solution?

Key Establishment Solutions (continued)

(1) Simplest solution is a network-wide shared key

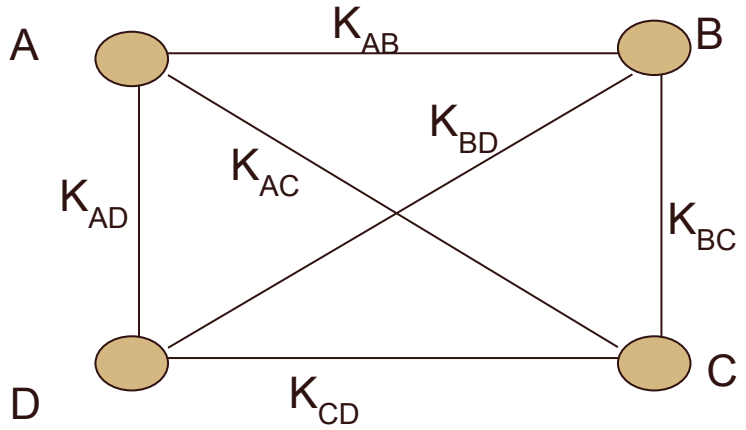
- **Problem**: if even a single node was compromised, the secret key would be revealed, and decryption of all network traffic would be possible

Key Establishment Solutions (continued)

(2) Point-to-Point security

- Need a key for every pair of nodes in an *n* node network.
- Trivial solution requires storing n – 1 keys at every node.
  - Not scalable on the space usage.

A-B	K_{AB}
A-C	K_{AC}
A-D	K_{AD}
B-C	K_{BC}
B-D	K_{BD}
C-D	K_{CD}



Secrecy and Authentication

We need cryptography as protection against eavesdropping, injection, and modification of packets

Trade-offs when incorporating cryptography into sensor networks:

- end-to-end cryptography achieves a high level of security but requires that keys be set up among all end points

- link-layer cryptography with a network-wide shared key simplifies key setup, but intermediate nodes might eavesdrop or alter messages

Key Establishment Solutions (continued)

(3) Bootstrapping keys using a **trusted base station**

- Each node needs to share only a single key with the base station and set up keys with other nodes through the base station

- The base station becomes a **single point of failure**
  - Utilize tamper-resistant packaging for the base station, reducing the threat of physical attack
  - Most existing work assumes base station is safe

Key Establishment Solutions (continued)

(4) Random-key pre-distribution protocol

- Large pool of symmetric keys is chosen
- Random subset of the pool is distributed to each sensor node
- To communicate, two nodes search their pools for a common key
  - If they find one, they use it to establish a session key
  - Not every pair of nodes shares a common key, but if the key-establishment probability is sufficiently high, nodes can securely communicate with sufficiently many nodes to obtain a connected network

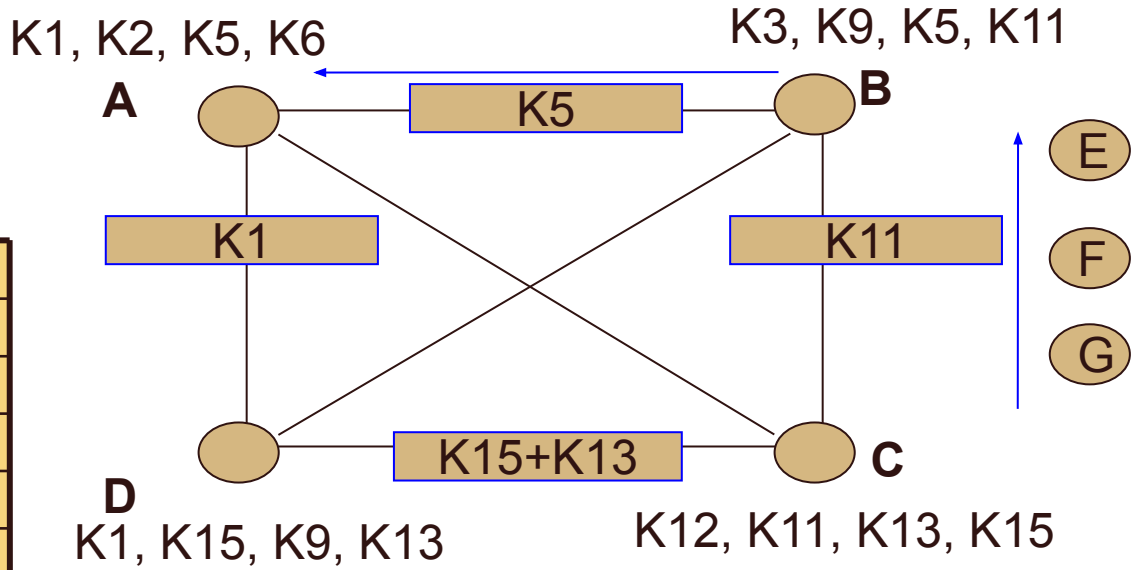
- No need to include a central trusted base station

- **Disadvantage**:
  - Attackers who compromised sufficiently many nodes could also reconstruct the complete key pool and break the scheme

Random key pre-distribution

Pool of Keys
K1, K2, K3, K4, K5, K6,
K7, K8, K9, K10, K11,
K12, K13, K14, K15

A	K1, K2, K5, K6
B	K3, K9, K5, K11
C	K12, K11, K13, K15
D	K1, K15, K9, K13
E	K10, K4, K5, K8, K7
F	K3, K5, K7, K9, K15
G	K1, K5, K9, K13

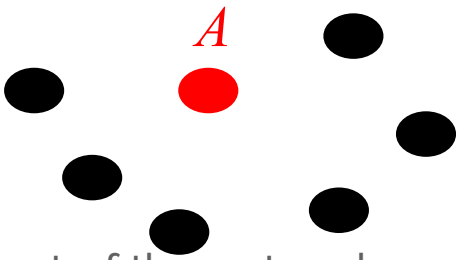


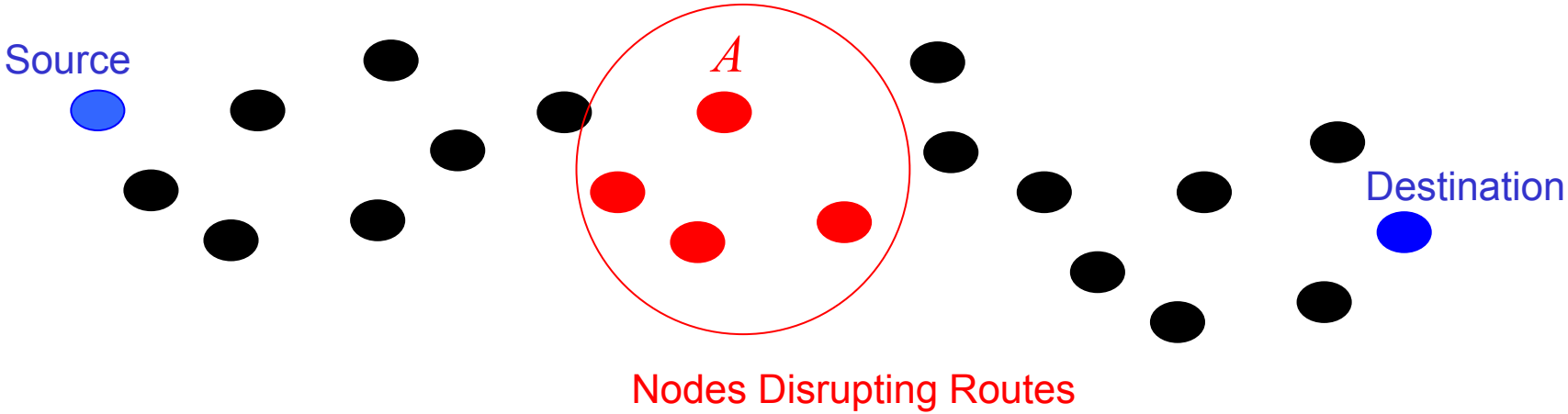
Denial of Service (DoS)

Can exhaust the medium by sending noise continuously.

- Simple form: Radio jamming
- Sophisticated form: Transmit while a neighbour is also transmitting or continuously generating a request-to-send signal

One strategically adversary can generally disable a dense part of the network.

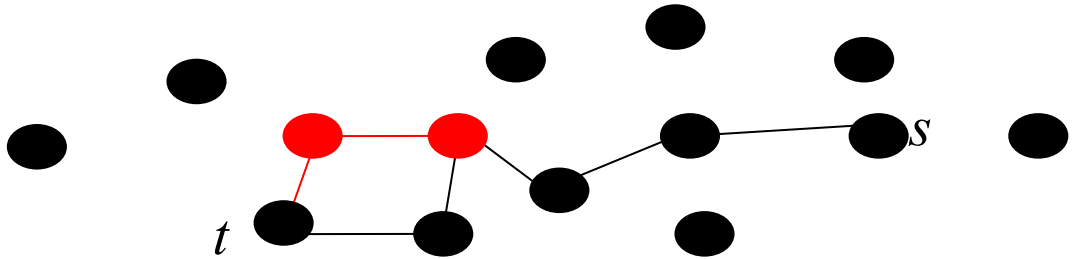




Denial of Service (DoS)

Packet rerouting – also known as data plane attacks.

Attacker reveals paths but does not forward data along these paths.



Robustness to Denial of Service

Possible solution (when the jamming affects only a portion of the network):

- Detect the jamming
- Map the affected region
- Route around the jammed area

802.11 standard uses Access Control Lists for admission control.

- If MAC address not in the list, then the node is denied access.

Secure Routing

Proper routing and forwarding are essential for communication in sensor networks

Injection attacks

- Transmit malicious routing information into the network resulting in routing inconsistencies
- Authentication might guard against injection attacks, but some routing protocols are vulnerable.

Sensor network routing protocols are particularly susceptible to node-capture attacks

- Compromise of a single node could be enough to take over the entire network or prevent any communication within it

Secure Routing (Resilience to Node Capture)

In traditional computing, **physical security is often taken for granted**

Sensor nodes, by contrast, are likely to be placed in **open locations**

- Attacker might capture sensor nodes
- Extract cryptographic secrets
- Modify programs/Replace them with malicious nodes

Tamper-resistant packaging may be one defense, but it’s expensive

Secure Routing (Algorithmic Solutions to Node Capture)

Attempt to build networks that operate correctly even in the presence of nodes that might behave in an arbitrarily malicious way

- Replicate state across the network and use majority voting to detect inconsistencies
- Gather redundant views of the environment and crosscheck them for consistency

Most challenging problems in sensor network security

- We are far from a complete solution!!

More!

Heartbleed - affected nearly every web server:
https://www.youtube.com/watch?v=8oI_laHhGjE

Solution?
UPDATE YOUR MACHINES

Meltdown/Spectre:
https://www.youtube.com/watch?v=bs0xswK0eZk

Solution?
UPDATE YOUR MACHINES