EECS 388: Embedded Systems

11. Security (Part 2)

Heechul Yun

Agenda

- Security attributes
- Threat model
- Software security
- **Information flow**
- **Encryption**
- **Digital signature and hashing**
- **SSL/TLS**

Information Flow

- Many security properties concern the FLOW of information between different principals in a system.
  - Confidentiality: preventing secret → attacker
  - Integrity: preventing attacker → system
- Information flow security is the study of how such flows affect the security and privacy properties of a system.

KU THE UNIVERSITY OF KANSAS

Example 1: Illegal Information Flow?

```
int patient_id; // initialized to the
                // patient's unique identifier
void take_reading() {
   float reading = read_from_sensor();

   display(reading);

   send(network_socket, hospital_server,
        reading, patient_id);

   return;
}
```

Example 2: Illegal Information Flow?

```
int patient_id; // initialized to the
                // patient’s unique identifier
long cipher_text;

struct secret_key_s {
  long key_part1; long key_part2;
}; // struct type storing 128-bit AES key

struct secret_key_s secret_key; // shared key

void take_reading() {
  float reading = read_from_sensor();

  display(reading);

  enc_AES(&secret_key, reading, &cipher_text);

  send_enc(network_socket, hospital_server,
           cipher_text, patient_id);

  return;
}
```

Example 3: Illegal Information Flow?

```
int patient_id; // initialized to the
                // patient's unique identifier
int patient_pwd; // stored patient password

float stored_readings[100];

void show_readings() {
  int input_pwd = read_input(); // prompt user for
                                // password and read it
  if (input_pwd == patient_pwd) // check password
    display(&stored_readings);
  else
    display_error_mesg();

  return;
}
```

The fact that you failed to login
Leak some information about
Your password

Limiting Password Attempts

- To limit information leakage, most today's devices disable them after a few failed attempts.

Invasive Attack

```
int patient_id; // initialized to the
                // p
int patient_pwd;

float stored_readings[100];

void show_readings() {
  int input_pwd = read_input(); // prompt user for
                                // password and read it
  if (input_pwd == patient_pwd) // check password
    display(&stored_readings);
  else
    display_error_mesg();

  return;
}
```

What if the attacker is capable of directly reading from the memory?

Secure Storage and Hashing

```
int patient_id; // initialized to the
                // patient's unique identifier
int patient_pwd_hash = read_from_secure_storage(…)

float stored_readings[100];

void show_readings() {
  int input_pwd = read_input(); // prompt user for
                                // password and read it
  if (hash(input_pwd) == patient_pwd_hash) password
    display(&stored_readings);
  else
    display_error_mesg();

  return;
}
```

Invasive Attack

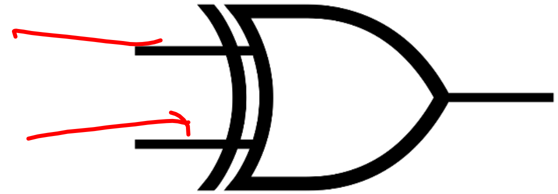
```c
int patient_id; // initialized to the
                // patient's unique identifier
long cipher_text;

struct secret_key_s {
  long key_part1; long key_part2;
}; // struct type storing 128-bit AES key

struct secret_key_s secret_key; // shared key

void take_reading() {
  float reading = read_from_sensor();

  display(reading);

  enc_AES(&secret_key, reading, &cipher_text);

  send_enc(network_socket, hospital_server,
           cipher_text, patient_id);

  return;
}
```

What if the attacker is capable of directly reading from the memory?

Basic Cryptography

- Symmetric (shared key) crypto
  - XOR encryption (one-time pad)
  - DES (56 bit key)
  - AES (up to 256bit key)
- Asymmetric (public-key) crypto
  - RSA
- Digital signature and secure hashing
  - SHA-256

XOR



NPUT		OUTPUT
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

XOR Encryption

Assume M (a message) and K (a key) are n-bit words. Let the cipher text be

$$C = M \oplus K.$$

To decode C,

$$C \oplus K = (M \oplus K) \oplus K = M \oplus (K \oplus K) = M \oplus 0 = M.$$

This uses the facts that exclusive OR ($\oplus$) is associative and commutative, that $B \oplus B = 0$ for any B, and that $B \oplus 0 = B$ for any B.

KU THE UNIVERSITY OF KANSAS

Slide source: Edward A. Lee and Prabal Dutta (UCB)

XOR Encryption

Assume M (a message) and K (a key) are n-bit words. Let the cipher text be

$$C = M \oplus K.$$

Note that without knowing K, C contains no information about M. This is because for any given M, we can generate any C by choosing the right K,

$$K = C \oplus M.$$

This is because

$$C = M \oplus K = M \oplus (C \oplus M) = C \oplus (M \oplus M) = C.$$

Slide source: Edward A. Lee and Prabal Dutta (UCB)

Example

- Encryption

```
        01010111 01101001 01101011 01101001 ← M: message (“Wiki”)
XOR     11110011 11110011 11110011 11110011 ← K: repeat key (11110011)
-------------------------------------------------
=       10100100 10011010 10011000 10011010 ← C: encrypted message
```

- Decryption

```
        10100100 10011010 10011000 10011010 ← C: encrypted message
XOR     11110011 11110011 11110011 11110011 ← K: repeat key
-------------------------------------------------
=       01010111 01101001 01101011 01101001 ← M: message (“Wiki”)
```

https://en.wikipedia.org/wiki/XOR_cipher

XOR Encryption

A key K should be used only once! Suppose

$$C_1 = M_1 \oplus K$$

and

$$C_2 = M_2 \oplus K.$$

Then

$$C_1 \oplus C_2 = M_1 \oplus M_2,$$

which reveals *some* information about M_1 and M_2. Specifically, if you determine one, you can determine the other.

How?

Slide source: Edward A. Lee and Prabal Dutta (UCB)

Example

- Recovering the key from M and C

```
        01010111 01101001 01101011 01101001 ← M: message (“Wiki”)
XOR     10100100 10011010 10011000 10011010 ← C: encrypted message
-----------------------------------------------
=       11110011 11110011 11110011 11110011 ← K: repeat key (11110011)
```

- Pros and Cons of XOR Encryption
  – Inexpensive
  – Insecure when key is used repeatedly and/or part of the message is known

Symmetric (Shared Key) Cryptography

- Block cipher uses more elaborate algorithms so that key size and message size don't need to be the same.
- Data Encryption Standard (DES) – mid 1970s.
- Advanced Encryption Standard (AES) – 2001 Based on a cryptographic scheme called Rijndael proposed by Joan Daemen and Vincent Rijmen, two researchers from Belgium. AES uses a message block length of 128 bits and three different key lengths of 128, 192, and 256 bits.

KU THE UNIVERSITY OF KANSAS

Asymmetric (Public Key) Cryptography

- Each participant has two keys, a public and a private one.
- A message is encrypted with the public key.
- The message can only be decrypted with the private key.
- Public and private keys match via clever algorithms.
- Relies on a *one-way function,* easy to compute, hard to reverse without knowing a (private) key.

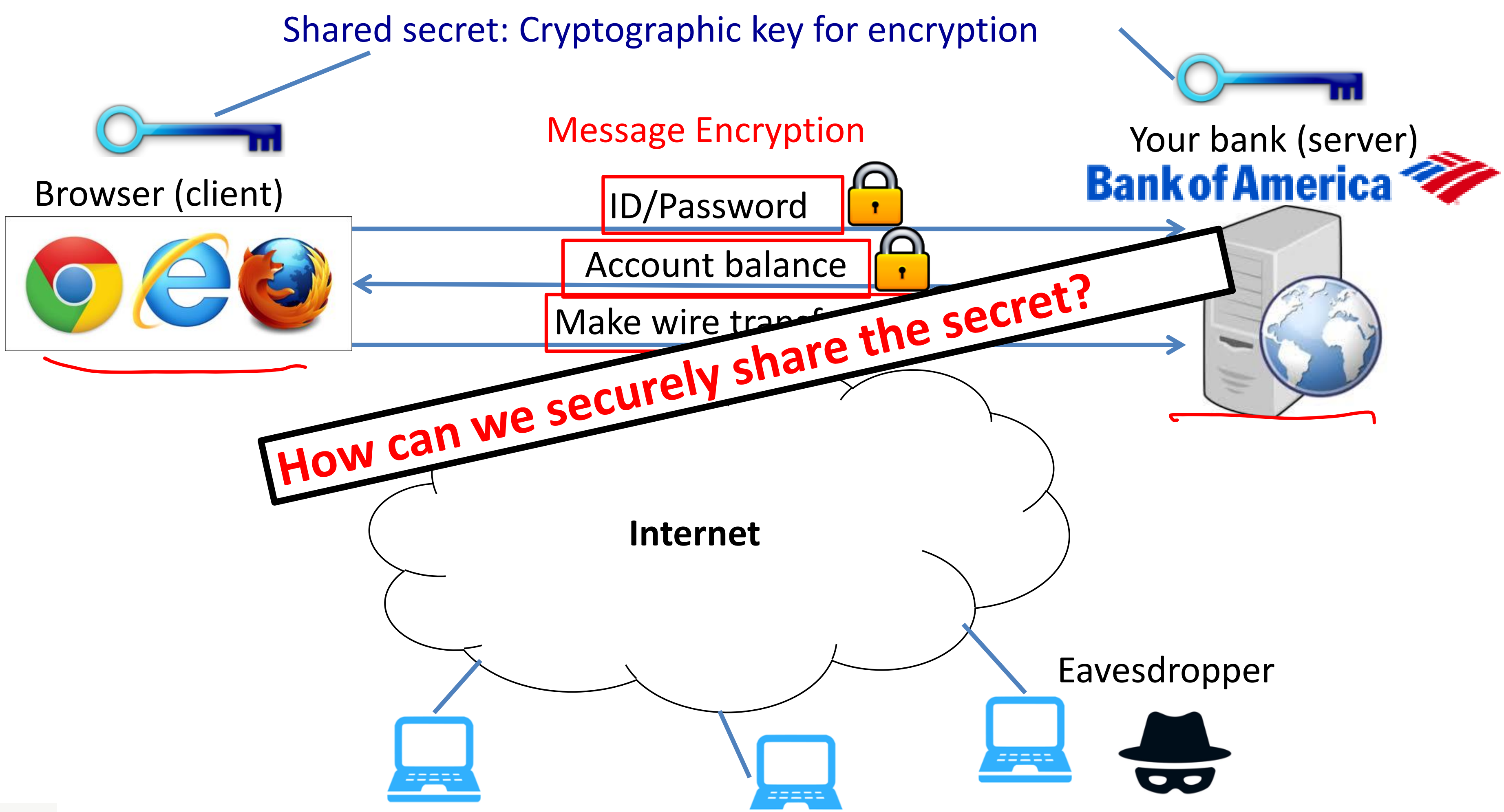
KU THE UNIVERSITY OF KANSAS

SSL/TLS

- Secure Socket Layer/Transport Layer Security
  - Widely used for web servers on the Internet
  - Provides:
    - Authentication
    - Confidentiality and integrity of communication

HTTPS = HTTP over SSL/TLS

Intro to SSL/TLS Based on Certificates

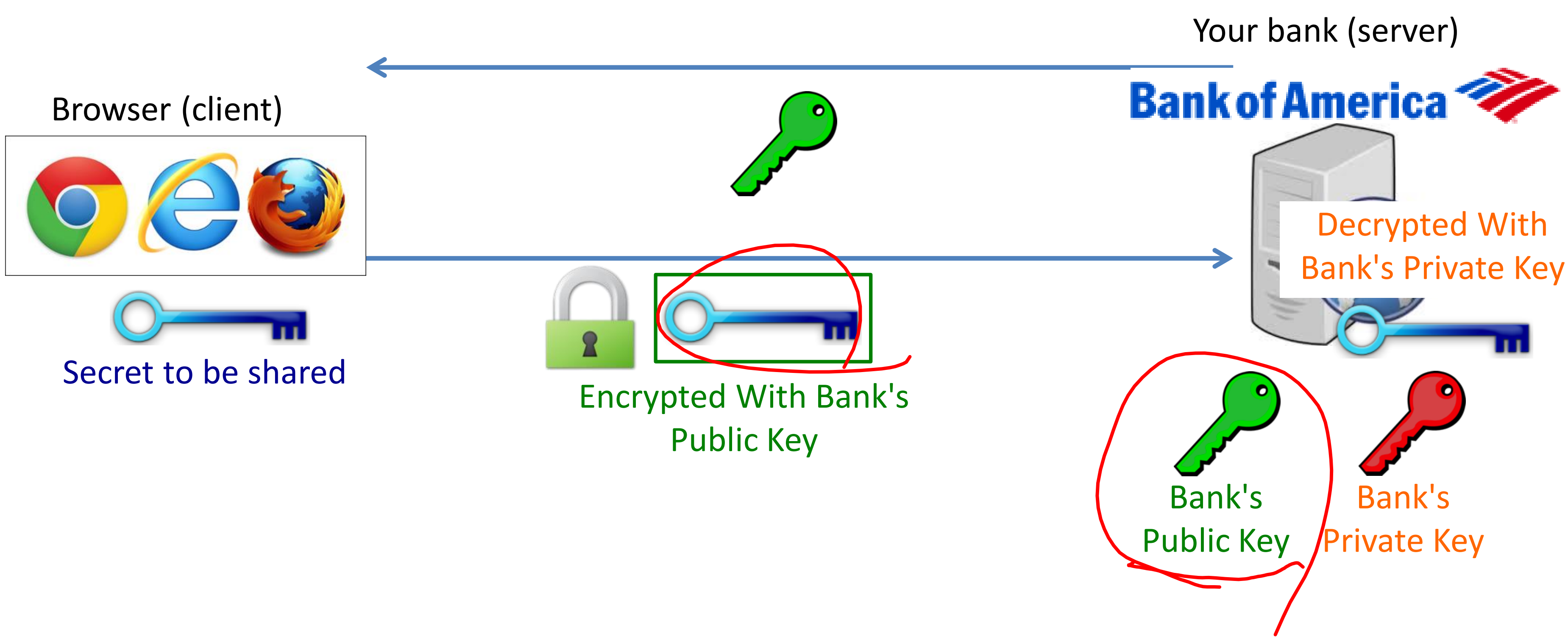


KU THE UNIVERSITY OF KANSAS

Slide source: Hokeun Kim and E. A. Lee (UCB)

Intro to SSL/TLS Based on Certificates

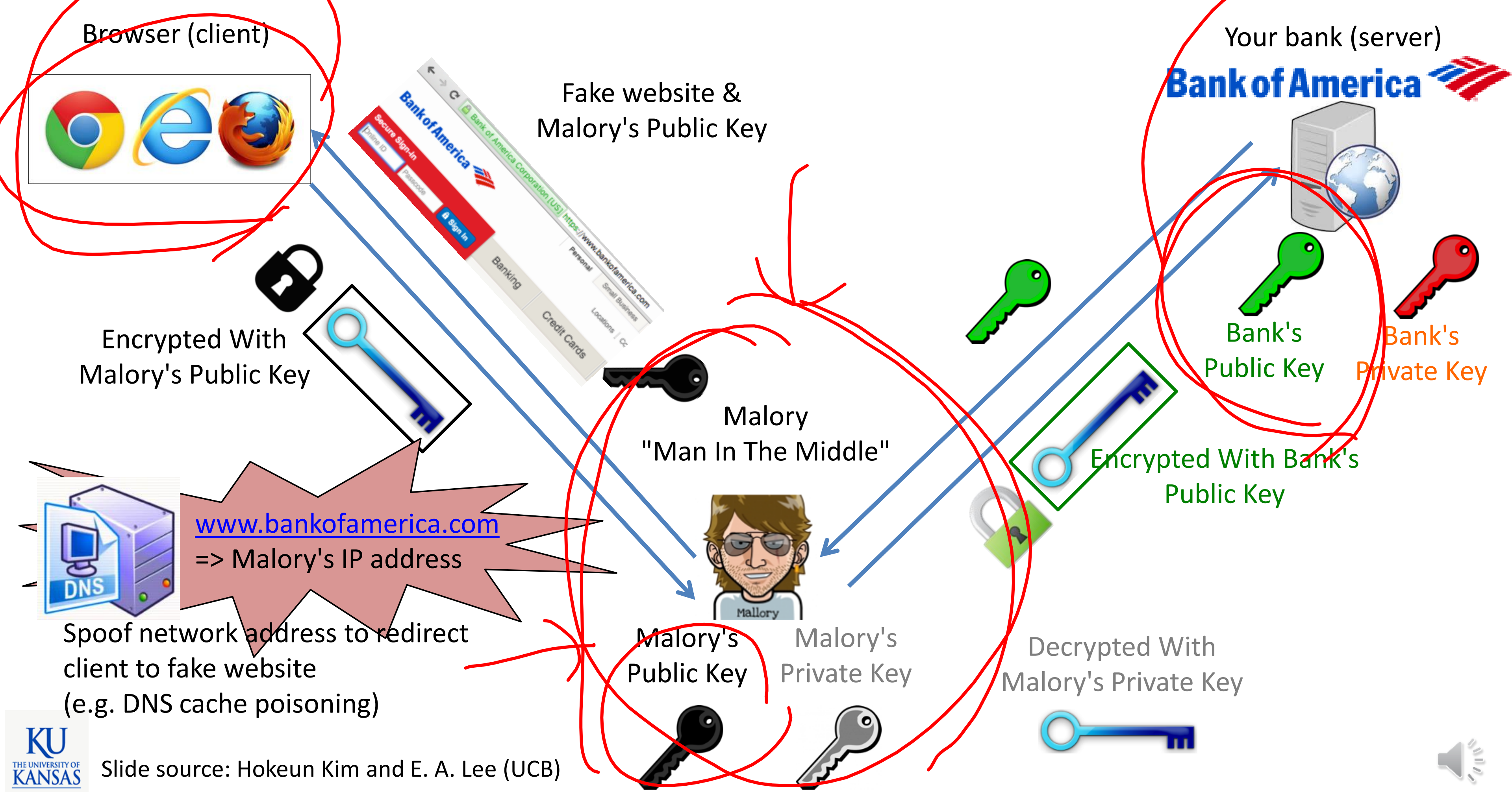
- Public key cryptography (e.g., RSA)



Slide source: Hokeun Kim and E. A. Lee (UCB)

Intro to SSL/TLS Based on Certificates

- However, even with public key cryptography…



Signing a Message

- Each participant has two keys, a public and a private one.
- A message is encrypted with the *private* key and both the message and its encryption are sent.
- The encrypted part can be decrypted with the *public* key. If it matches the plaintext message, the signature is valid.

Intro to SSL/TLS Based on Certificates

A (Digital) Certificate (Proof of Public Key's Authenticity)

- Name of certificate authority (CA)

  Symantec. digicert

- **www.bankofamerica.com**

  Bank of America

- Bank's **public key**

```
00:d6:  e:3c:c3:b7:b0:0e:c2:a3:0b:a8:7e:d6:
72:7  .e4:29:d0:2b:9b:30:ec:cc:d2:05:c1:0e:17:
41:e0:e3:58:42:98:95:73:06:13:8a:41:99:fb:69:
79:70:95:3e:77:69:c0:70:31:01:6f:fa:22:02:8e:
```

- Additional Information: validity period, etc.

Signed (encrypted)* with issuer (CA)'s Private key

- Digital Signature

```
7c:29:18:b4:57:8e:f5:bf:ec:bc:14:ea:ac:b9:3e:ad:13:dc:
3a:77:e7:7a:ab:3b:23:46:46:4a:2d:ee:7a:d0:
43:d6:17:d1:c6:86:ff:a0:b5:33:c4:de:ec:d8:
a6:cb:0f:02:a8:22:c7:fb:98:b1:75:61:b1:9d:
```

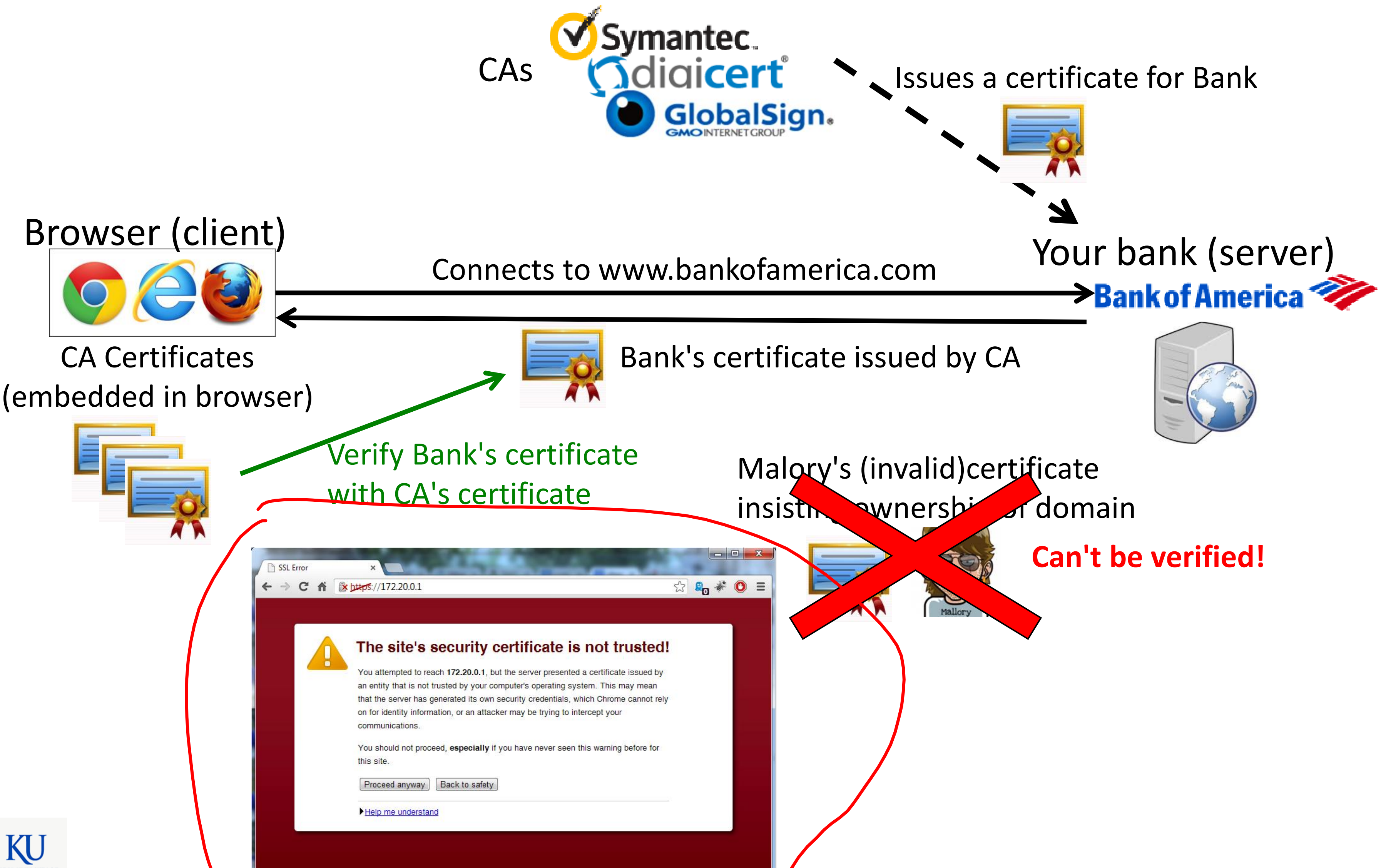
Can only be decrypted (verified) with issuer (CA)'s matching public key!

Actually the hash of data is encrypted (signed), and the result of decryption is also hash

KU THE UNIVERSITY OF KANSAS

Slide source: Hokeun Kim and E. A. Lee (UCB)

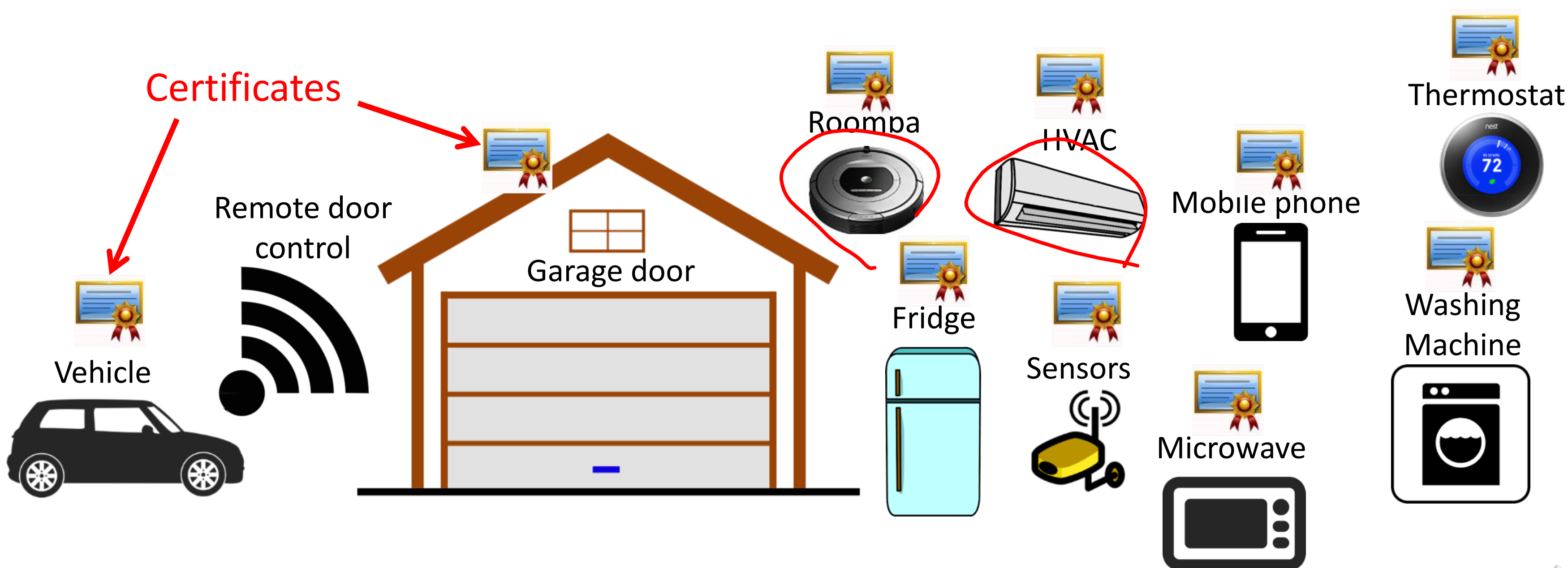
Intro to SSL/TLS Based on Certificates



Slide source: Hokeun Kim and E. A. Lee (UCB)

Issues with Using SSL/TLS for IoT

- Overhead for **resource-constrained devices**
  - Energy/computation overhead for public key crypto, communication bandwidth, memory, etc.
- Limited support **one-to-many communication**
  - Connections are 1-to-1 (server/client model)



Slide source: Hokeun Kim and E. A. Lee (UCB)

Issues with Using SSL/TLS for IoT

- Management overhead of certificates
  - If you use commercial certificate authorities (CAs)

GlobalSign GMO INTERNET GROUP

DomainSSL	1 Year	2 Years	3 Years	
Add-on Options:	only $249	only $448	only $613	Buy Now

digicert

3 Years	$139 per year	(Best Value - You Save $108)	BUY OR RENEW
2 Years	$157 per year	(You Save $36)	BUY OR RENEW
1 Year	$175		BUY OR RENEW

Quotes from www.digicert.com

- Company Validation
  … First, we will verify that the company requesting a certificate is in good standing …
- Domain Validation
  … can include emails or phone calls to the contact listed in a domain's whois record …

  - Alternative: free & automated CA   Let's Encrypt
    - Overhead for managing domains to get certificates

Slide source: Hokeun Kim and E. A. Lee (UCB)

Summary

- Security used to be an after thought (if any)
- In networked embedded systems (a.k.a. IoT) security is a first-class concern
- Embedded systems security are even harder than desktop/server security because of:
  - Diversity (no standard os, hardware, runtime, …)
  - Resource constraints (performance, energy, memory space, …)
  - The prevalent use of C (insecure language)
- Read chapter 17, take security courses…

Acknowledgements

- Security slides are based on the materials developed by
  – Edward A. Lee and Prabal Dutta (UCB) for EECS149/249A