ECE 547/647: Security Engineering

Meeting 7: Public Key Crypto and Smart-Grid

Readings

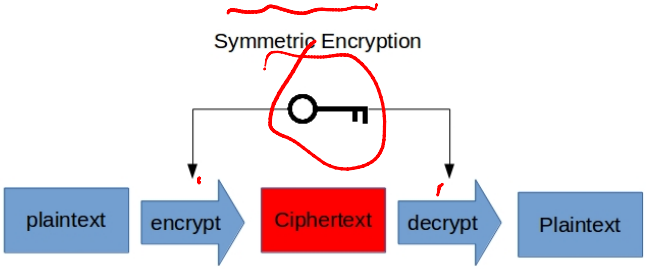
- Smart Grid Security
- Who Controls the Off-Switch?

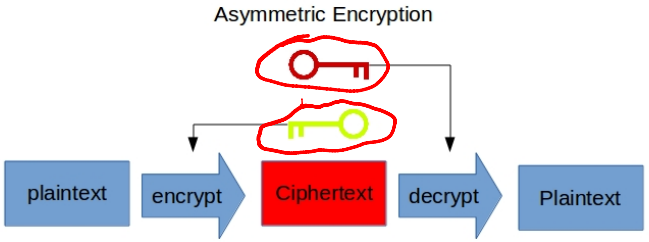
Asymmetric Cryptography, aka Public Key

- Alice and Bob protocols
- Key exchange – Diffie-Hellman
- Modular arithmetic
- RSA
  - How it works
  - Weaknesses
- Digital Signature
- Public Key Infrastructure (PKI)
- Certificates
- Tools

Symmetric vs Asymmetric

- Symmetric codes: Use same key for encryption and decryption
- Asymmetric codes: Use different keys for encryption and decryption





Symmetric vs Asymmetric

Symmetric disadvantages

- Need additional cryptographic operations for distributing the key
- Key update becomes difficult
- If the key is exposed, cryptographic algorithm fails

Asymmetric disadvantages

- More complex than symmetric
- Slower to implement
- Fewer options

Key exchange

- Two parties want to securely communicate with each other
- There needs to be a key exchange mechanism between them
- We talked about public key where each party shares their public key
- There are other key exchange methods available
- Diffie-Hellman key exchange (DH) : a method of securely exchanging cryptographic keys over a public channel
- One of the first public key protocols

Diffie-Hellman Key Exchange

- Why does DH work?

Merkle, Hellman and Diffie, 1976
The Guardian,

Diffie-Hellman Key Exchange

- How can two parties achieve message confidentiality who have no prior shared secret and no secure channel to exchange one?
  - Select a suitably large prime number p and a base $g \in \mathbb{Z}_p^* \ (2 \ \leq g \leq p-2)$, which can be made public, Both parties agree to use these two numbers
  - A generates x and B generates y, both random numbers out of $\{1, \ldots, p-2\}$

$$A \rightarrow B \ transmission: \quad g^x \bmod p$$
$$B \rightarrow A \ transmission: \quad g^y \bmod p$$

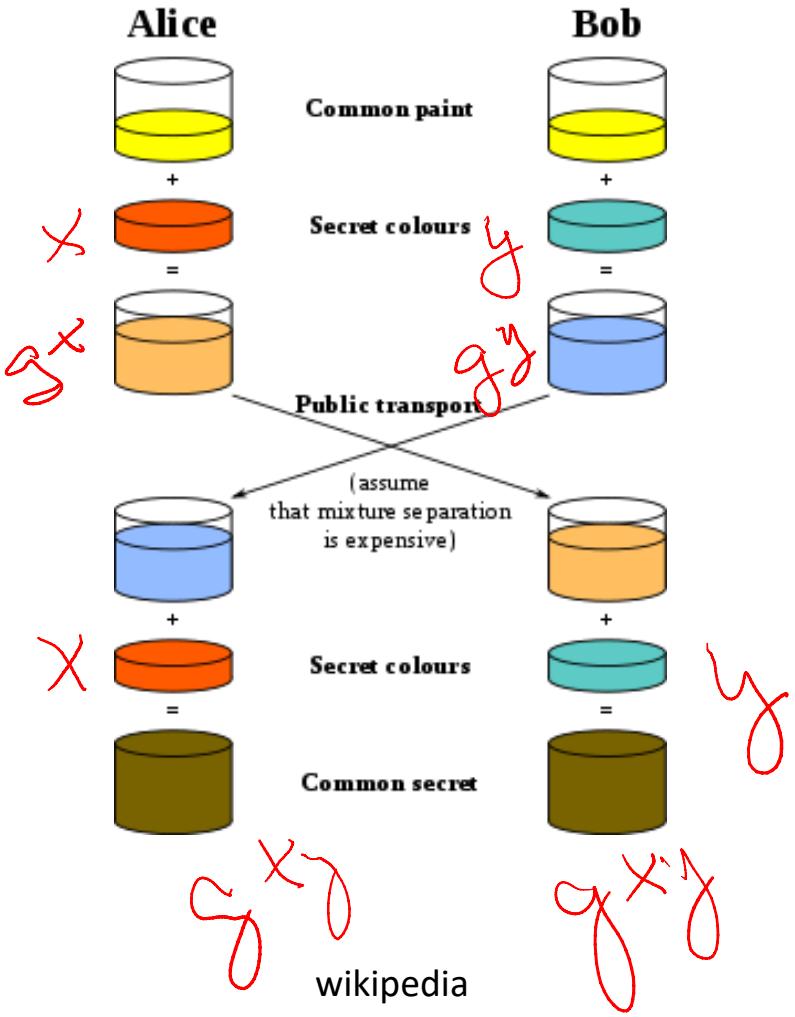
  - Now both can form $(g^x)^y = (g^y)^x$ and use a hash of it as a shared secret key

Diffie-Hellman Key Exchange

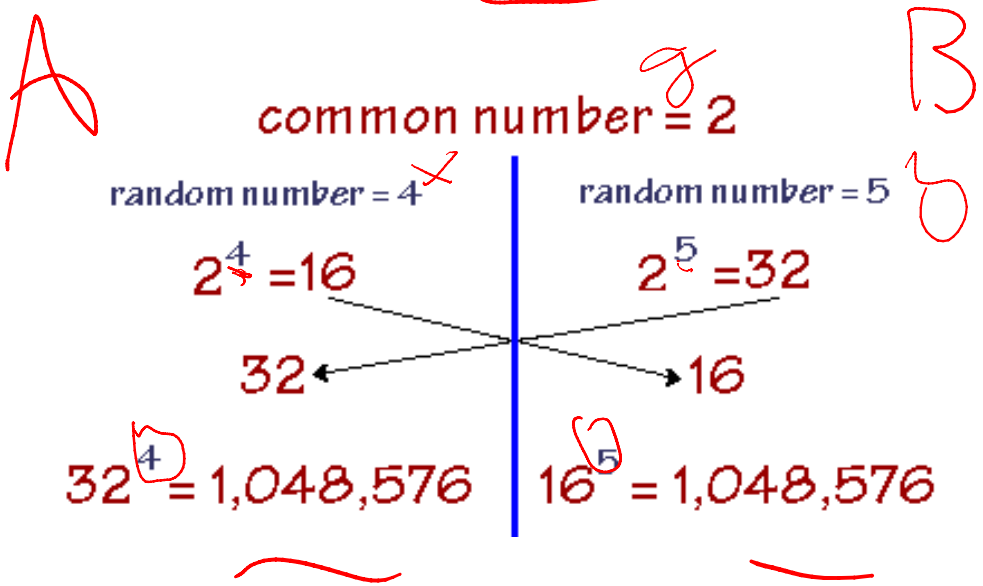
- The eavesdropper faces the Diffie-Hellman Problem of determining g^{xy} from g^x, g^y, and g, which is believed to be equally difficult to the Discrete Logarithm Problem of finding x from g^x and g in $\mathbb{Z}_p^*$

- This is infeasible if $p > 2^{1000}$ and $p - 1$ has a large prime factor

Diffie-Hellman analogy



DH example (but no mod so easy to break!)



Quiz 2

- What are two security challenges in Smart Grids?
  - 1
  - 2

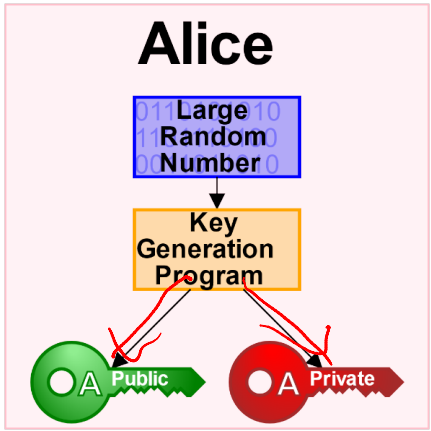
- What are two potential solutions to these challenges?
  - 1
  - 2

Public key cryptography

- A public key is defined in the context of asymmetric key
- Everyone has access to the public key and little effort is made to ensure its security
- A party can use public key to encrypt a message but cannot use the same key to decrypt the message.

Public Key Cryptography

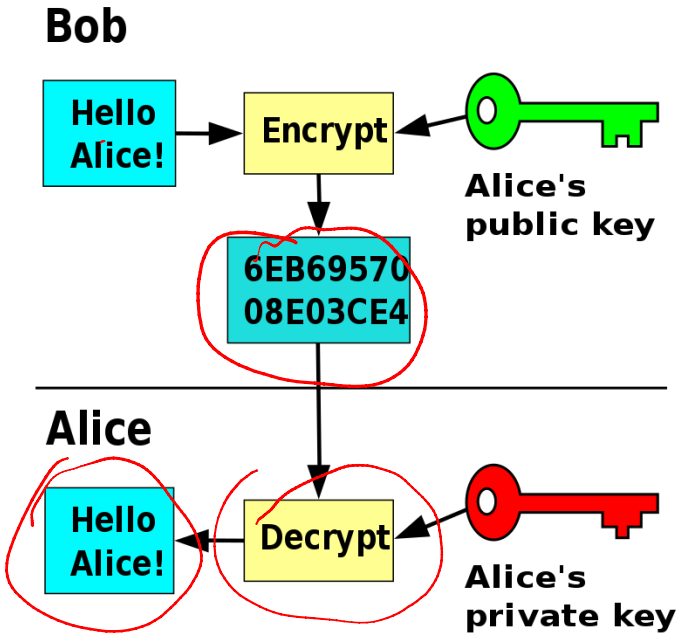
- Key distribution made easier
- Just make sure everyone has access to your public key
- A private key is associated with the public key that can be used to decrypt the message
- In asymmetric cryptographic algorithms, the private/public key pair are generated from a random number



Source: Wikipedia

Public Key Encryption

- Everyone has everyone else's public key
- If Bob wants to transmit a message to Alice, he encrypts the message with Alice's public key
- This way he makes sure that only Alice can decrypt the message using her private key
- An adversary cannot eavesdrop without access to the private key which is not distributed and therefore not vulnerable



Source: Wikipedia

Some terminology

- Plaintext: m
- Ciphertext: c
- Public Key: $c = K^{+}(m)$
- Private Key: m = $K^{-}(c)$
- Alice Keys: K_A^{+}, K_A^{-}
- Bob Keys: K_B^{+}, K_B^{-}

Requirements

- Requirements for a secure asymmetric cryptographic algorithm

1. Ability to generate K^+ and K^- from random number such that

$$K^+\left(K^-(m)\right) = m \qquad K^-\left(K^+(m)\right) = m$$

2. Given K^+ it should be computationally implausible to calculate K^- and vice versa

- Examples of public key (asymmetric) algorithms
  - RSA, ElGamal, Elliptic curves…
  - Post-Quantum methods: lattice, coding, hash, hyper-elliptic,…

Example quiz questions

- Why do we use the term **asymmetric** cryptography?

- What mathematical property does DH key exchange rely on?

Show what problem the attacker would need to solve to learn the secret key?

Discrete Log

Public key encryption

- PKE has 3 parts: key generation, encryption, decryption
- RSA is a very well known PKE, developed by three researchers:

  Ronald **R**ivest , Adi **S**hamir, Leonard **A**delman – 1977

RSA Algorithm Key Generation

- Generate one public key and one private key from random numbers
- Choose two prime numbers (usually really large) p, q
- Compute n=pq and z=(p-1)(q-1)
- Choose *e* (with *e<n*) that has no common factors with *z* (relatively prime)
- Choose *d* such that *ed-1* is exactly divisible by *z (e mod z = 1*) , ed-1=z*t
- Public key = K^+ = (n,e) , Private key = K^+ = (n,d) , both are a pair of numbers

RSA Encryption/Decryption

- A message can be considered a string of bytes, each byte represents a number between 0,255
- We encrypt each byte (m <n) separately
- Encrypt: $c = K^{+}(m) = m^{e} \ mod \ n$
- Decrypt: $\mathrm{m} = K^{-}(c) = c^{d} \ mod \ n$
- Example: bit pattern = 00000101 -> m = 5
- p = 5, q = 7, n = 35, z = 24, e=5, d=29, K^{+}=(35,5), K^{-}=(35,29)
- c = $m^{e} \ mod \ n$ = 10
- m = $c^{d} \ mod \ n$ = 5

Why RSA Works?

- We need to prove: m=$K^{-}(K^{+}(m))$

$K^{-}(K^{+}(m))$ = $K^{+}(m)^{d}\ mod\ n = \left(m^{e}\ mod\ n\right)^{d}$ mod n = $m^{d*e} mod\ n$

We know that d*e = z*t+1

So $K^{-}(K^{+}(m)) = m^{z*t+1} mod\ n = m^{(p-1)*(q-1)*t+1} mod\ n$

Fermat's little theorem

1- $m^{(p-1)*(q-1)*t+1}$ mod p = $m * (m^{(p-1)})^{(q-1)*t} = m * (1)^{(q-1)*t}$ = m

2- By Symmetry, $m^{(p-1)*(q-1)*t+1}$ mod q = m

Based on the Chinese remainder theorem

$m^{(p-1)*(q-1)*t+1}$ mod p*q = m (QED)

How Secure is RSA?

- We need to make sure that having access to the public key, it is computationally infeasible to get the private key
- If we know 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.
- We should check every factor of n for brute force attack
- factoring a big number is hard, no efficient algorithm exists, takes at least sqrt(n) tries! Non-feasible for large n values!
- Takes more than two years on a powerful cluster with hundreds of cores to factorize a 232-digit number
- a single core 2.2 GHz AMD Opteron processor with 2 GB RAM takes about 1500 years!

Weak points of RSA

- Getting the key pair can be computationally expensive
- How many clock cycles before we find two large enough prime numbers?
- Getting e is hard (finding a co-prime number)
- d is unique and finding it is hard

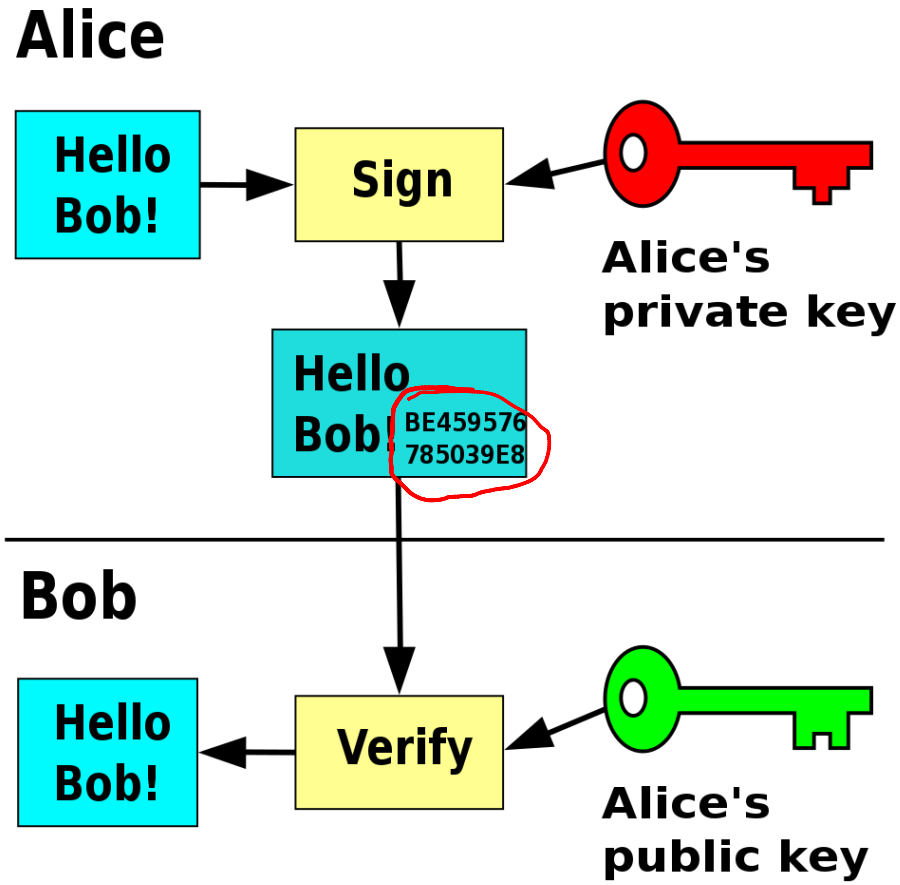
- Quantum computing (Shor's algorithm, 1994) could allow practical factoring of large prime numbers…

Weak points of RSA

- No Random Component -> vulnerable to chosen plaintext attack
  - Adversary has a plaintext ciphertext pair
  - Adversary encrypts similar plaintexts under the public key and tests the compares the ciphertexts and can draw conclusions based on the difference
- Product of two ciphertexts is equal to the encryption of the product of the two corresponding plaintexts
  - Adversary can perform a chosen ciphertext attack
  - Adversary asks the key holder to decrypt a non-suspicious ciphertext, using the decrypted message and previously known plaintext/ciphertext pairs, the adversary can find out the key

Digital Signature

- Bob receives a message from Alice, how does he make sure that it is Alice who has sent him the message?
- A digital signature refers to a set of algorithms and encryption protections used to determine the authenticity of a message, document, or software



Source: Wikipedia

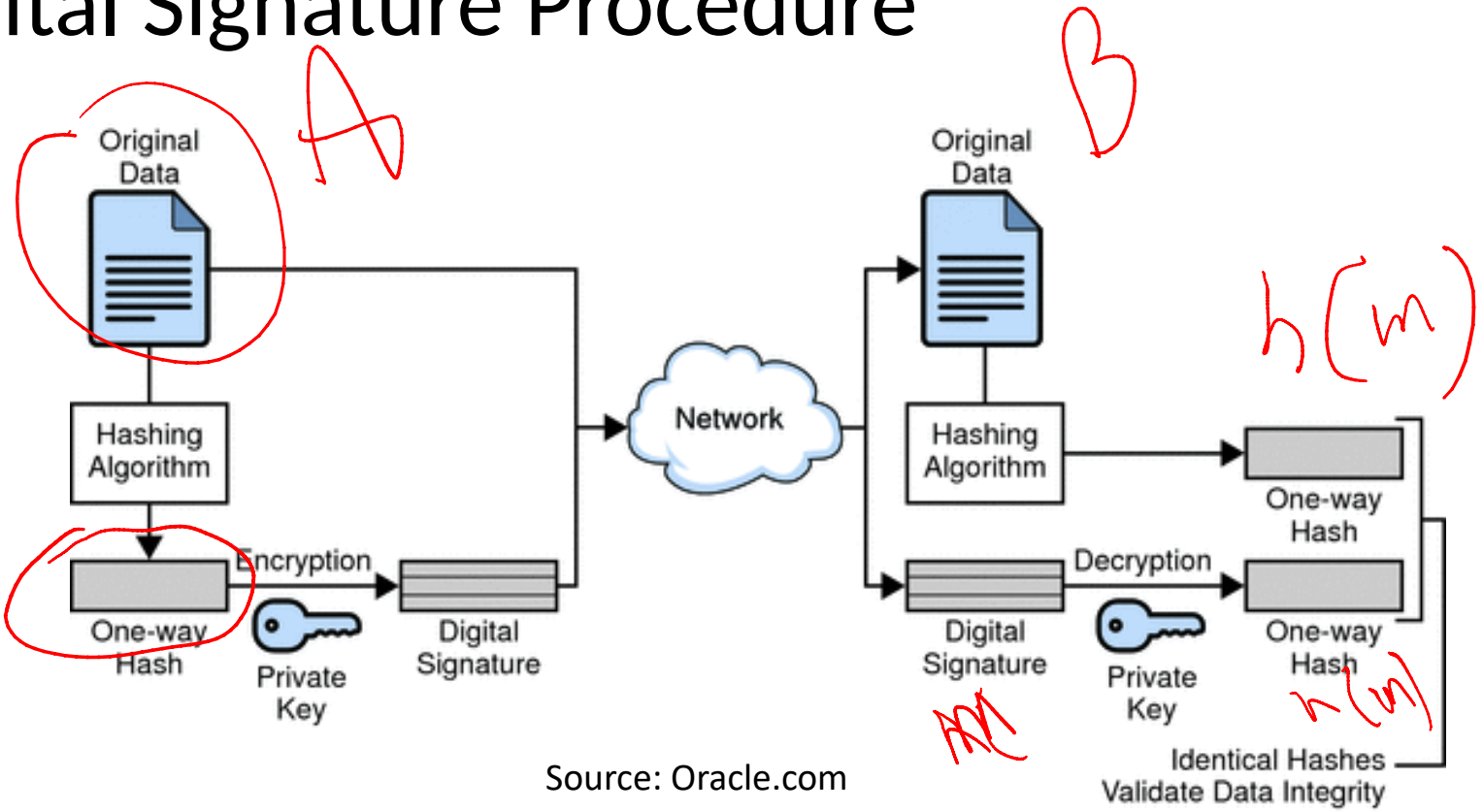
Digital Signature

- Alice transmits a message m to Bob, she also encrypts the message with her private key and adds that to the message as digital signature
- Bob receives msg m from Alice as well as the digital signature $K_A^-(m)$
- Bob verifies m signed by Alice by applying Alice's public key K_A^+ to $K_A^-(m)$ then checks if $K_A^+(K_A^-(m)) = m$
- if $K_A^+(K_A^-(m)) = m$, whoever signed m must have used Alice's private key.

Digital Signature Procedure

- Alice chooses an initial message (m)
- Alice hashes the message with a secure hash $h(m)$
- Alice encrypts the hashed message with her private key $K^{-}(h(m))$
- Digital signature to transmit: (m, $K^{-}(h(m))$)
- Bob receives the signature and decrypts the message $K^{+}(K^{-}(h(m)))$
- Bob also hashes the received message to get $h(m)$
- Bob compares the two messages above, if they are equal, the signature is verified
- Why hash?

Digital Signature Procedure



Source: Oracle.com

Another use of Digital Signature

- Using digital signature:
- Bob verifies that Alice signed the message m
- No one else signed the message or tampered with it!
- Non-repudiation
  - Bob can use the signature as a proof of Alice's participation

But wait! How to know whose public key?

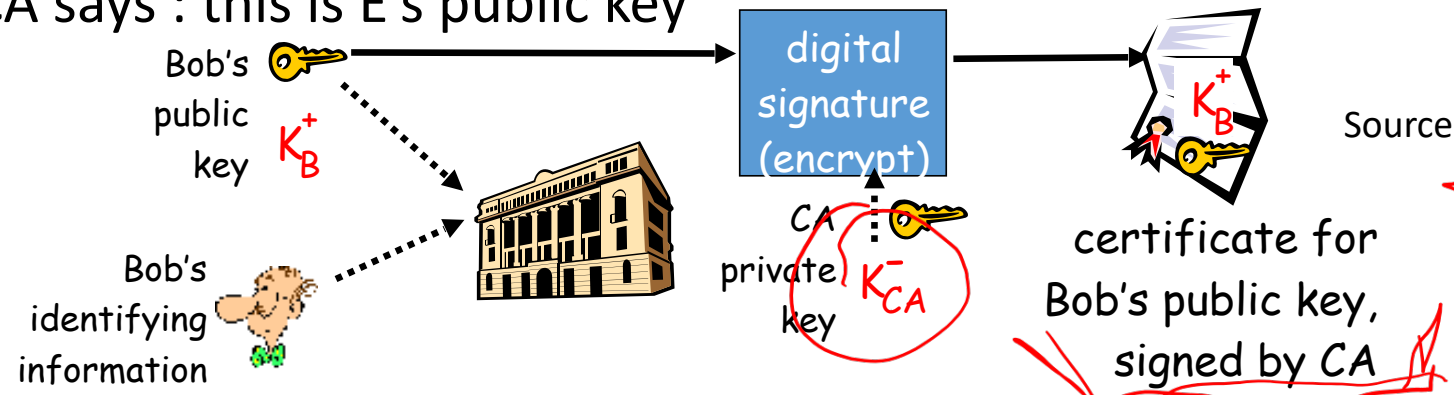
- Motivation: Trudy plays pizza prank on Bob
- Trudy creates e-mail order:
  Dear Pizza Store, Please deliver to me four pepperoni pizzas. Thank you, Bob
- Trudy signs order with her private key
- Trudy sends order to Pizza Store
- Trudy sends to Pizza Store her public key, but says it's Bob's public key.
- Pizza Store verifies signature; then delivers four pizzas to Bob.
- Bob doesn't even like Pepperoni

Public Key Infrastructure

- We need a secure and trusted third party to store the public key of people and transfer it to us when needed
- **A public key infrastructure (PKI)** is a set of roles, policies, hardware, software and procedures needed to create, manage, distribute, use, store and revoke digital certificates and manage public-key encryption.
- The purpose of a PKI is to facilitate the secure electronic transfer of information for a range of network activities such as e-commerce, internet banking and confidential email.

Certificate Authority

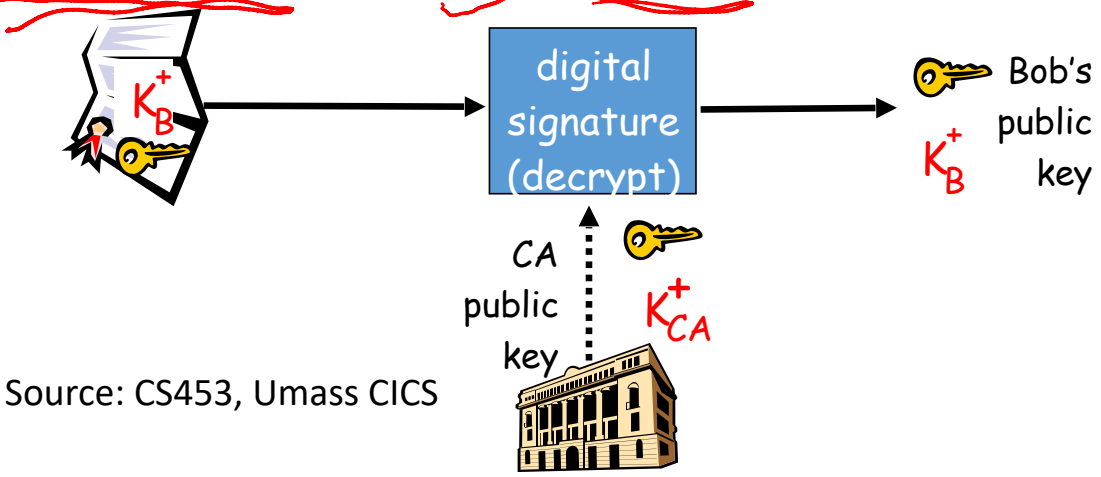
- **Certification authority (CA):** binds public key to particular entity, E.
- E (a person or company) registers its public key with CA.
- E provides a proof of identity to CA.
- CA creates certificate binding E to its public key.
- With the certificate containing E's public key digitally signed by CA , the CA says : this is E's public key



Source: CS453, Umass CICS

Certificate Authority

- When Alice wants Bob's public key:
- she gets Bob's certificate from the Bob himself or somewhere else
- apply CA's public key to Bob's certificate, get Bob's public key



Source: CS453, Umass CICS

Public-Key Infrastructure

- Such a **certification authority C** issues a digitally signed **public key certificate**

$$\text{Cert}_C(A) = \{A, K, T, L\}_{K_C^{-1}}$$

- in which C confirms that the public key K belongs to A starting at time T and that this confirmation is valid for the time interval L, and all this digitally signed with C's private signing K_C^{-1}
- Anyone who knows C's public key K_C from a trustworthy source can use it to verify the certificate $\text{Cert}_C(A)$ and obtain a trustworthy copy of A's key K_A in this way

Some Popular Unix Cryptography Tools

- **`ssh [user@]hostname [command]`** – Log in via encrypted link to remote machine (and if provided execute "command")
  - RSA or DSA signature is used to protect Diffie-Hellman session-key exchange and to identify machine or user
  - Various authentication mechanisms, e.g. remote machine will not ask for password, if user's private key (`~/.ssh/id_rsa`) fits one of the public keys listed in the home directory on the remote machine (`~/.ssh/authorized_keys2`)
  - Generate key pairs with `ssh-keygen`

Some Popular Unix Cryptography Tools

- **`psp, gpg`** – Offer both symmetric and asymmetric encryption, digital signing, and generation, verification, storage and management of public-key certificates in a form suitable for transmission via email

- **`openssl`** – Tool and library that implements numerous standard cryptographic primitives, including AES, X.509 certificates, and SSL-encrypted TCP connections

More Unix Cryptography Tools

- **`ssh [user@]hostname [command]`**
  - http://www.openssh.org
- **`psp, gpg`**
  - http://gnupg.org
- Distributed key directory:
  - http://www.cam.ac.uk.pgp.net/pgpnet/wwwkeys.html

Smart Grid Security, Khurana et al, IEEE Security and Privacy, 2010

- Intro to Smart Grid
- Challenges
  - Trust
  - Comms and Device security
  - Privacy
  - Management
    - Complexity
    - Scale
- Solutions
  - Requirements
  - Authentication and Encryption

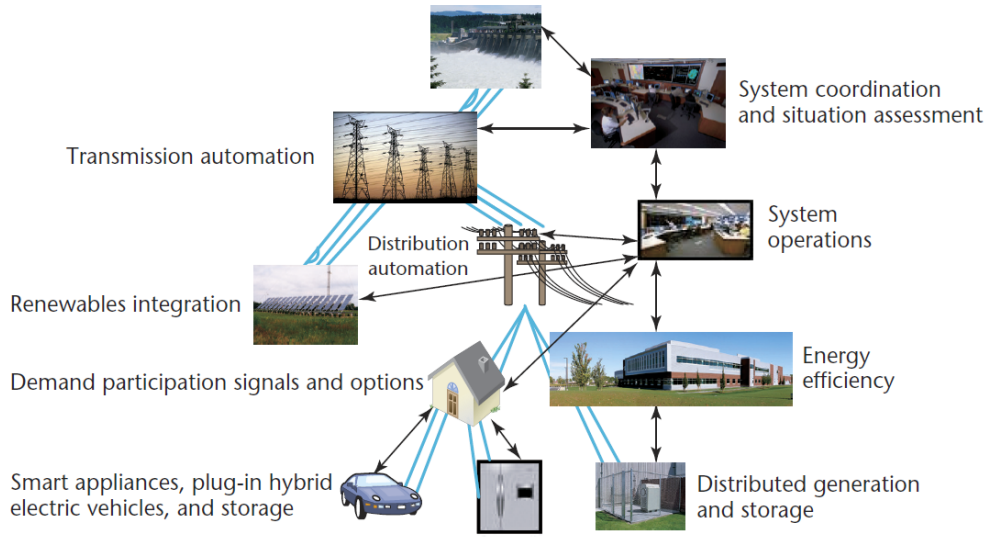


Figure 1. The smart grid's physical layers and communication and control systems. Smart-grid data availability places considerably more stringent demands on the communication and control system than traditional supervisory control and data acquisition (SCADA) systems do.

Key management

- Public Key infrastructure
- Key generation
- Certificates
- Human resources for IT management

Who controls the off switch?

Anderson and Fuloria, 2010, 1st Conf on Smart Grid.

- Additional reading for assignment
- Our textbook author
- Several clever threat models
- Privacy!

Assignment (to be formalized on Moodle)

- The Voting paper and both Smart Grid papers are both from 2010. What is different in 2020?
- Pick two issues from each paper to show what has changed and what has not in the last 10 years.
  - Issues can include:
    - New Security Challenges
      - New threat models
      - Scaling of systems
    - New solutions
      - Computing and communication technology
      - New algorithms, networking, crypto
- Write 300 words on each topic (4 x 300 = 1200). Include a reference section, being sure to formally cite the 2 original papers and at least 4 additional papers/sources (probably within the last 2-3 years)