Introduction
–draft; to be updated at the end of the week–

1 Course Overview

1.1 Organization

In short: Security properties and primitives, classical crypto, public key systems, toolbox, and real world applications (and some implementations). We shall discuss some advanced protocols and privacy later in the course, along with some higher-level thoughts on policy.

Large parts of this course will be centered around various cryptographic primitives. We shall ask ourselves: What can we do with it? What are its properties? How does it work? (And, sometimes, how is it implemented?)

1.2 What this course is not

This course is broad rather than deep. We won't go too deep into the mathematics underlying most of the primitives, and shall only briefly discuss cryptanalysis. We will also not talk about security mechanisms for computer and network devices and applications such as firewalls, operating system access controls, detecting software security holes, or dealing with web security vulnerabilities.

2 The State of Security

2.1 Breaches

Massive security breaches are disclosed almost daily, from identity theft, ransomware, and surveillance, to nation-state cyberwarfare, huge denial-of-service attacks, and industrial espionage. Added to all of this is large-scale misuse of personal data by legitimate companies.

Examples: Target, Marriott, Apple vs FBI, Stuxnet, Heartbleed, BGP issues, Anonymization (Netflix), WannaCry, etc. Nice link: `https://informationisbeautiful.net/`

`visualizations/worlds-biggest-data-breaches-hacks/`

2.2 Attackers

It is very important to think about who the attackers are (the term of art is adversary, or, in the industry, threat actor).

- John "Captain Crunch" Draper (Phreaking)
- Kevin "Condor" Mitnik (LA Buses, corporations)
- Julian "Mendax" Assange (Wikileaks; Pentagon, US Navy, Citibank, NASA, Lockheed-Martin)
- Albert "soupnazi" Gonzalez (ShadowCrew, ATM cards)
- The Russian Business Network (ISP, cybercrime, crime in general, cybercrime-as-a-service)
- PLA Unit 61398 (APT, state-sponsored industrial espionage)
- The US Government (at various levels, especially the NSA and CIA)
- Edward Snowden
- Bureau 121 and No. 91 Office (NK; Bangladesh Bank, Sony Pictures, etc.)

2.3 Defenses and the role of Cryptography

Cryptography is to information security as locks are to personal security. Both are clever mechanisms that can be analyzed in isolation and can be effective when used in suitable contexts, but comprise only a small part of the security picture.

Note: Information security as a whole would involve protection against data damage, theft of intellectual property, surveillance, unauthorized actions, etc.

In the real world, we may achieve security using:

- Prevention: Physical barriers, locks, encryption, firewalls, etc.
- Detection: Audits, checks and balances.
- Legal means: Laws, sanctions.
- Concealment: Camouflage, steganography.

3 Security Principles

Consider an on-line banking web site: Think about the interests of the customer, the bank, and possible intruders. Can the bank trust its customers? What about the other way around?

We usually consider something (ironically) called the CIA triad: Confidentiality, Integrity, and Availability.

C access is granted only to authorized individuals (e.g., using encryption, access control, authentication)

I data is not modified or lost in an unauthorized manner (e.g., backups, correcting codes, etc.)

A data is available to authorized entities when required (e.g., protections and redundancies)

3.1 Threats and countermeasures

Some risks and possible countermeasures:

- Eavesdropping on private conversations: encryption.
- Unauthorized use of a computer: passwords, physical security.
- Unwanted email: spam filters.
- Unintentional data corruption: checksums and backups.
- Denial of service: redundancy, isolation.
- Breach of contract: nonrepudiable signatures.
- Data corruption: access controls, cryptographic hash functions.
- Disclosure of confidential data: access controls, encryption, physical security.

Discuss: alteration (man-in-the-middle), DDoS, masquerading, correlation and traceback.

In general, there is no such thing as absolute security. The best we can do is to optimize the tradeoff between the cost of security and losses from breaches. This may involve reducing vulnerabilities, or reducing the value (or increasing the cost) of a successful attack, or heavily penalizing failed attacks, etc.

4 Symmetric Cryptography

Let's say we want to transmit a message M securely (what does that mean?) from Alice to Bob. One way to achieve this would be the use of a symmetric or private-key (or single-key) cryptosystem. This is a pair of efficiently-computable functions E and D such that:

$E(k, m)$ encrypts plaintext message m using key k to produce a ciphertext c.
$D(k, c)$ decrypts ciphertext c using k to produce a message m.

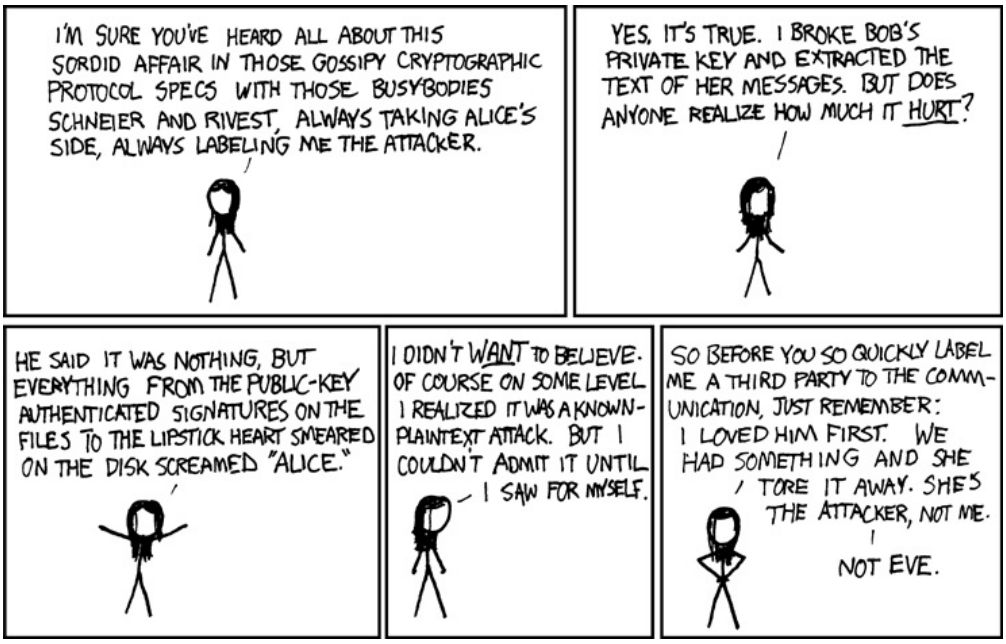
Correctness: $D(k, E(k, m)) = m$ for all keys k and all messages m.
Security: Given $c = E(k, m)$, it is hard to find m without knowing k.

4.1 A theoretical example

So, our "protocol", if we can call it that, would look something like this:

1. Alice and Bob share a secret key k.
2. Alice computes $c = E(k, m)$ and sends c to Bob.
3. Bob received c' and computes $m' = D(k, c')$ and assumes $m = m'$

We assume that an eavesdropper Eve learns nothing except for c during the protocol; also, Eve is a passive eavesdropper who can only read c but can not modify it. Further, we assume that the channel is perfect, so $c' = c$. The real world is not so perfect, so we must ask what happens when $c' \neq c$?



4.2 Requirements

What do we require of E, D, and the computing environment?

- E and D can be computed quickly, and that they are correct and secure (as defined earlier)
- Given c, it is hard to find m without also knowing k.
- k is not initially known to Eve.
- Eve can guess k with at most negligible success probability. (k must be chosen randomly from a large key space.)
- Alice and Bob successfully keep k secret. (Their computers have not been compromised; Eve can't find k on their computers even if she is a legitimate user, etc.)
- Eve can't obtain k in other ways, e.g., by social engineering, using binoculars to watch Alice or Bob's keyboard, etc.
- More?

4.3 Formalization

A symmetric cryptosystem consists of

- a set M of plaintext messages,
- a set C of ciphertexts,
- a set K of keys,
- an encryption function $E : K \times M \to C$
- a decryption function $D : K \times C \to M$

We often write $E_k(m) = E(k, m)$ and $D_k(c) = D(k, c)$.

Now, recall our old definitions of correctness and security. Let's discuss these notions:

- What's a feasible amount of computational time and storage?
- Finding m: Always? Sometimes? Some bits? Some function of m? What is we can modify m?
- What does "hardness" mean here? What does it mean to not know k?

- Auxiliary / a priori knowledge of keys. How do we choose k? (We need a source of randomness that is not available to Eve.)

You'll want to think about complexity and difficulty of implementation. Note that "k remains secret" is a naive claim of security; it is both too strong and too weak: Eve at always brute-force guess k, and you could have a cryptosystem where k remains private but m can be found easily.

Some attack scenarios (we'll discuss these later):

- Ciphertext only: Eve knows only c and tries to recover m.
- Known plaintext: Eve knows c and a set of plaintext-ciphertext pairs $(m_1, c_1), \ldots, (m_n, c_n)$ $c \notin \{c_1, \ldots, c_n\}$. Eve tries to recover m. (This might happen if the plaintext is revealed publicly later, such as in an auction, or be guessable, like an email header, etc.)
- Chosen plaintext
- Chosen ciphertext
- Both CPA and CCA
- Adaptive CPA, CCA

Think about when CPA and CCA would actually occur in real life. For now (though we might edit this later), we'll make do with this definition: A cryptosystem is computationally secure relative to a notion of compromise if, for all probabilistic polynomial-time algorithms A, when given as input the "security parameter" n and all of the information available to Eve, the algorithm succeeds in compromising the cryptosystem with success probability that is negligible in n.

Next week