Foundations of Cryptography 2021-22
Lecture 1

Some history, Example: CSAM by Apple, The Sentinel Problem (Entropy and Identification) *

Moni Naor

1 History of Cryptography

We discussed some of the key events of cryptography. We mentioned Kerckhoffs's principle (1883) which states that a cryptosystem should be secure *even if everything about the system, except the key, is public knowledge*[1]. We also mentioned Shannon's 1949 paper (not to be confused with his 1948 paper establishing Information Theory). See link in the homepage of the course. There was of course a lot of classified work and there are fascinating stories about breaking cryptosystems during World War II, most famously the Enigma. One comprehensive Source about the history of cryptography is David Kahan's "The Codebreakers" published in 1966. But to a large extent cryptography was not investigated by academic researchers, there were only a few scientific papers published in the open literature on the subject.

In the mid 1970's several important developments occurred that changed the . They need for cryptography due to the development of computer and communication systems, the advent of Complexity Theory and probably Zeitgeist "spirit of the times", in particular lack of trust in authority. Key events:

- Publication of Diffie and Hellman paper "New Directions in Cryptography", 1976 introduced many new ideas including public-key cryptography.
- Publication of the RSA paper - first trapdoor and signatures in 1978.
- DES - Data Encryption Standard, for symmetric key encryption, developed by IBM and made a US standard in 1977. This gave a respectable option for people wishing to use symmetric key encryption, which was suitable to the computational power available.

While the traditional setup of cryptography dealt with two parties - Alice and bob - who talk and an adversary Eve who listens (Eavesdrops), modern cryptography has considered more involved models for more diverse tasks. A possible definition is that it deals with methods for maintaining

*These notes summarize the material covered in class, usually skipping proofs, details, examples and so forth, and possibly adding some remarks, or pointers. In the interest of brevity, most references and credits were omitted.

[1]This means that in most circumstances a cryptographic system must have a (random) secret key.

the secrecy, integrity and functionality in computer and communication system in light of an adversarial threat.

In this course we will emphasize a rigorous approach to specification of security. A fundamental idea of cryptography is to use the computational intractability of some problems in order to build secure systems. This is an idea that is applied in most, but not all, construction we will see in the course.

we discussed several examples of why cryptography is "in the news" and relevant to everyday life. Examples include (but are limited to):

- Internet security and privacy
- Security of mobile communication
- Ctypto-currencies, block-chain etc.
- Voting system
- Apple's proposal for preventing CSAM being used on its cloud.

2 CSAM

Apple has recently (Summer 2021) proposed (and quickly withdrew) a system whose goal is to make it harder to use the apple cloud in order to store image generated by child secular abuse. There is a more or less fixed database of around 200K images and the goal is alert The host (Apple) if a user tries to store more than a certain number of images (or derived images) from the database.

The proposed system uses many techniques that we will study in the course. For instance, Private Set Intersection (PSI). Two parties, each having a subset, want to learn the intersection of the subsets without revealing more information than this. Defining what it means to not learn more information that the intersection is a non-trivial task.

The variant that is relevant in this case is where the host (Apple) has a more or less fixed subset that it cannot know explicitly. Only the host learns the result and the host learns only if the intersection is larger than a certain threshold.

One data structure used is *Cuckoo Hashing*, which is a dictionary where each item may be stored in one of two locations (Determined by public hash functions). This reduces the problem to something closer to a comparison, i.e. for each image the user has, the amount of work performed is proportional to constant number of images, independent of the CSAM Database size. . This in turn done using a Diffie-Hellman encryption and using the random-self-reduction properties of the scheme (this is something we will cover later in the course)..

Threshold Secret Sharing: To achieve the threshold property requirement of the PSI the system uses so called *secret sharing schemes*. In such a scheme A dealer would like to split a secret $s \in \{0,1\}^\ell$ between n users, each receiving a share, so that any t of the users may reconstruct the

secret from their shares, but fewer than t users learn nothing regarding s. Unlike almost any other task, this can be performed with information theoretic notions of security.

We saw an extremely simple 2-out-n scheme that does not even require identities (See [2]. The notion was introduced by Adi Shamir who gave the famous construction based on polynomial interpolation over a finite field [3].

3 Specifying the Security of a system

To define security of a system must specify:

- What constitute a failure of the system.
- The power of the adversary in term of
  - Computational power.
  - Access to the system - i.e. which parts are available to the adversary.
- What it means to break the system. E.g. if this is a game, when does the adversary win.

4 The Sentinel Problem and Entropy

We will define in the next class several notions of entropy. In general entropy measures some sort of information content a random variable has and depending on what we are trying to measure affects the definition.

Let X be random variable over alphabet Γ with distribution P_X. The (Shannon) entropy of X is

$$H_1(X) = -\sum_{x \in \Gamma} P_X(x) \log P_X(x)$$

Where we take $0 \log 0$ to be 0.

The Shannon entropy represents how much we can compress X (expected length to encode X under the best code). Examples:

If $X = 0$ (i.e. it is constant) then $H_1(x) = 0$ and the only case where $H_1(x) = 0$ is when X is constant. All other cases $H_1(x) > 0$

If $\Gamma = \{0, 1\}$ and $\text{Prob}[X = 0] = p$ and $\text{Prob}[X = 1] = 1 - p$, then

$$H_1(X) = -p \log p + (1 - p) \log(1 - p) \equiv H(p)$$

If $\Gamma = \{0, 1\}^n$ and X is uniformly distributed, then

$$H_1(X) = -\sum_{x \in \{0,1\}^n} 1/2^n \log 1/2^n = 2^n/2^n \cdot n = n$$

and this is when the entropy is maximized.

For passwords the Shannon Entropy may not be such a great property for distribution of passwords in the sense that it may be pretty large and yet pretty bad as password distribution. Consider the distribution where with probability $1/2$ the result is 0^n and with probability $1/2$ it is uniform over $\{0,1\}^n$.

Pre homework: compute the Shannon entropy of this distribution.

Instead we considered the *Min Entropy* of a distributions as a more relevant parameter:

$$H_\infty(X) = \max_{x\in\Gamma} -\log p_X x.$$

That is, if x the most frequent element the $-\log p_X x$

Finally we mention the *Collision entropy.* sometimes just called "Renyi entropy",

$$\mathrm{H}_2(X) = -\log \sum_{x\in\Gamma} p_X x^2 = -\log P(X=Y),$$

where X and Y are iid.

The single guard problem is: Alice and Bob share a setup, not known to Eve. At some point Alice wants to send an 'Approve' message to Bob, a one-time identification event. The power the adversary has is that Eve may inject any message at any point in time. The required properties are:

Completeness: if Eve does not interfere and Alice wants to approve then Bob accepts (note that there are no requirements if she does interfere)

Soundness: if Alice does not approve, then no matter what Eve does, the probability that Bob accepts is at most some ϵ.

We argued that it is necessary to have setup and that the system may not be perfect in terms of soundness. i.e. there is a probability $\epsilon > 0$ that Eve might succeed. In particular, if ℓ bits are sent in the identification phase, then $\epsilon \geq 2^{-\ell}$, since Eve can simply guess a message that will make Bob approve.

4.1 The Two Sentinels Problem

What happens if there are two guards that share a setup and Eve can control one of them? This means that the setup is public. The proposed protocol is for Alice to choose a random $x \in \{0,1\}^n$ and puts as the public set up information (or public-key) $y = f(x)$ for a function $f\colon \{0,1\}^n \mapsto \{0,1\}^n$. To 'approve' Alice sends x and the guard compute $f(x)$ and compares it to the y that is in the public-key. The guard approve if they are equal, and goes into permanent reject if they are not equal.

What are the requirements from f? It should be easy to compute and hard to reverse. That is, there is a poly-time TM M that given $x \in \{0,1\}^n$ outputs $f(x)$. On the other hand, for any

Probabilistic poly-time TM, given y that was generated by choosing $x \in_R \{0,1\}^n$ and computing $f(x)$, the probability that M out an inverse of y (i.e. it output is in $f^{-1}(y)$ is bounded by a negligible function in n

5 Models and Side Channels

An important issue that we will not emphasize in the course is side-channel attacks, that is an adversary that gets some information about the "guts" of the player, e.g. keys are leaked vi EM radiation, or measuring acoustics (the noise a processor makes) or timing, precisely how long things take to compute. This is a very important family of attacks, but we will usually make the assumption that the honest players are shielded.

References

[1] W. Diffie and M. Hellman, New Directions in Cryptography, IEEE Trans. on Information Theory, Volume: 22, Issue: 6, Nov 1976.

[2] Ilan Komargodski, Moni Naor and Eylon Yogev, *How to Share a Secret, Infinitely*, IEEE Trans. on Information Theory, Volume: 64, Issue: 6, Nov 2018, pp. 4179–4190.

[3] Adi Shamir, *How to Share a Secret* Communications of the ACM, Volume 22, Issue 11, Nov. 1979, pp 612-–613.