Foundations of Cryptography 2019/20
Lecture 1

A bit history, Primary Secondary Resolver Proofs systems, Entropy and Identification [*]

Moni Naor

1 History of Cryptography

We discussed some of the key events of cryptography. 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*. 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.

In the mid 1970's several important developments occurred. 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". Key events:

- Publication of Diffie and Hellman paper "New Directions in Cryptography", 1976 introduced many new ideas including public-key cryptography.
- Publication of RSA paper - first trapdoor and signatures. 1978
- DES - Data Encryption Standard, for symmetric key encryption, developed by IBM and made a US standard in 1977.

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 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. To define security of a system must specify:

[*] 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.

- What constitute a failure of the system.
- The power of the adversary
  - computational
  - access to the system
- what it means to break the system.

2 Primary Secondary Resolver Membership Proof Systems

As an example of a fairly involved system we considered Primary-Secondary-Resolver Membership Proof Systems (PSR for short): A PSR system is a 3-party protocol, where we have a Primary, which is a trusted party which commits to a set of members and their values $(name, value)$, then generates public and secret keys in order for the Secondary (ies; there an be several of them), provers with knowledge of both keys, and Resolvers, verifiers who only know the public key, to engage in interactive proof session regarding elements in the universe and their values. That is the resolver issue and query name and expects to get back the value or the response that the name is not in the database.

The motivation for such systems is for constructing a secure Domain Name System (DNSSEC) that does not reveal any unnecessary information to its clients. We require our systems to be *complete*, so honest executions will result in correct conclusions by the resolvers, *sound*, so malicious secondaries cannot cheat resolvers, and *zero-knowledge*, so resolvers will not learn additional information about elements they did not query explicitly. Providing proofs of (positive) membership is easy, as the primary can simply precompute signatures over all the members of the set. Providing proofs of non-membership, i.e. a denial-of-existence mechanism, is trickier and is the main issue in constructing PSR systems.

We outlined how such a system may look if we use a VPRF - a verifiable pseudorandom function as well as better known primitives such as digital signatures.

The pedagogical goal of discussing these PSR systems was to introduce advanced various notions

For more on PSR see `http://www.wisdom.weizmann.ac.il/~naor/PAPERS/psr_abs.html`

3 The Sentinel Problem and Entropy

We defined 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 know to Eve. At some point Alice wants to send an 'Approve' message to Bob, a one-time identification. Eve may inject any message at any point in time. The properties are: (i) Completeness - if Eve does not interfere and Alice wants to approve then Bob accepts (note that there are no requirements if she does interfere) (ii) 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. $\epsilon > 0$.

References

[1] John von Neumann, *Various techniques used in connection with random digits*, National Bureau of Standards Applied Math Series 12: 36-38, 1951. See `https://dornsifecms.usc.edu/assets/sites/520/docs/VonNeumann-ams12p36-38.pdf`