Lecture 10

FEBRUARY 21

Instructor: Fang Song

Disclaimer. Draft note. No guarantee on completeness nor soundness. Read with caution, and shoot me an email at fsong@pdx.edu for corrections/comments (they are always welcome!)

Logistics. Statistics. Supplement reading: BS more examples. Check resource page frequently. HW 4 out.
Last time. Private-key crypto recap.
Today. Public-key revolution.

1 Key distribution and management

We've been deferring a critical question for a long time:

How do the users share a *secret* key in the first place?

We can think of arranging secure *in-person meetings* to distribute secret keys from time to time. Alternatively, users may be able to use some trusted courier service. In both settings, we are essentially assuming existence of some secure channels available.

You may wonder, if a secure channel is present, why bother with any other tools from private-key crypto? Well the point here is that these forms of secure channels are either only available for a limited period of time, or are so costly and inefficient. Hence, while government and military agencies could afford such a solution, it is completely not feasible at a large-scale, e.g. on the Internet. For example, if there are N users on the Internet, to enable pair-wise secure communication, we would need to arrange $\Omega(N^2)$ meetings to distribute secret keys. Meanwhile, storing and managing a large number of keys by the users is difficult and prone to attacks.

The concerns above in principle can be addressed in a "closed"-system, where there is a well-defined population of users and they are willing to (and are capable of) following the same policies for distributing and storing keys. However, there is another issue in "open" systems which is actually more common once told. Consider using encryption to send credit-card information to an Internet merchant to complete a transaction for the very first time. In such transient interactions, one may not be aware of the other's existence until the time they want to communicate securely.

To summarize, there are at least three issues regarding the use of private-key cryptography:

i) how to distribute secret keys?

ii) how to store and manage large number of keys?

iii) how to handle dynamic interactions in open systems?

1.1 A partial solution: key distribution centers

FS NOTE: Draw KDC diagram

Assumptions: trust on KDC and secure channels between each user and the KDC. Channels between users don't have to be secure. Note that **trust** is an assumption, although slightly different from the other assumptions we are more familiar with (e.g., computational capability).

Advantages of KDC

- Each user needs to store only one long-term key (shared with the KDC). Sessions keys are short-term and are erased once a communication session concludes.
- Adding a new user amounts to setting up a key between new user and KDC.

Limitations of KDC

- Much workload on KDC.
- KDC is a single point of failure: if KDC is down (system error or intentional attacks), the entire system is unavailable or completely compromised.

2 Public-key Revolution

We still do not have a solution to distribute secret keys without invoking any private channels at all. In fact, it seems inevitable that some form of secure channel must be available. This is a common belief in the long history of cryptography (i.e. secret writing),which apparently needs no further justification. It is until very recent about half a century ago that people started to challenge this belief.

Here is my short, incomplete and perhaps biased version of the story at the dawn of public-key cryptography.

In the public domain, Ralph Merkle was probably the first to propose a brand new approach to key distribution. In Fall 1974, Merkle was then an undergrad at UC Berkeley, and he wrote in his project proposal [1] of a computer security course that

> "it might seem intuitively obvious that if two people have never had the opportunity to prearrange an encryption method, then they will be unable to communicate securely over an insecure channel... I believe it is false."

However, his proposal drew little interest from the professor who responded "not good enough". Merkle later dropped the class and continued working on it. Encouraged by another faculty member at Berkeley, he submitted a draft to the Communications of the ACM in August of 1975, but only to get comments as follows:

> "I am sorry to have to inform you that the paper is not in the main stream of present cryptography thinking and I would not recommend that it be published in the Communications of the ACM."
>
> "Experience shows that it is extremely dangerous to transmit key information in the clear."

[1] The original CS244 project proposal from Fall of 1974 (7 page PDF) `http://www.merkle.com/1974/FirstCS244projectProposal.pdf`. A two-page version `http://www.merkle.com/1974/SecondCS244projectProposal.pdf`.

No doubt, his paper got rejected. This only confirmed Merkle that no one had previously investigated this approach. After a long delay, his paper finally got published [Mer78] in 1978.

In essence what Merkle proposed was a protocol for two parties to exchange a secret key over a public (insecure) communication channel by using a publicly accessible hash function $\mathcal{O}$ (i.e. in modern term, it works in the random-oracle model). He showed that

- honest parties will agree on a key with N queries to $\mathcal{O}$.
- any eavesdropper has to spend $\Omega(N^2)$ queries to $\mathcal{O}$ to leran the key.

He conjectured that "it might be possible to obtain a protocol where breaking is exponentially harder than using them". But no concrete candidates were given.

History often shows the same pattern indicating the beginning of a *paradigm shift*. There were other people at the same time (and more to come) who envisioned innovative approaches to cryptography beside Merkle. The most influential were the two visionaries at Stanford then, Diffie and Hellman.

They realized the need for a revolution in cryptography that goes beyond the conventional domain of intelligence and military applications to a public environment such as commercial applications due to the "development of cheap digital hardware". They envisioned a new type of cryptography inspired by the "asymmetric" phenomenon in the physical world. Basically they imagined a magic box (we call them *trapdoor one-way permutations* these days), which is a collection of permutations $\{F_k\}$ which is easy to compute $F_k(x)$ but hard to invert $F_k^{-1}(y)$. But for each k, if some secret information $sk(k)$ (a *trapdoor*) is known then inverting becomes easy $x = F^{-1}(sk(k), y)$. With this, any can encrypt in a *public* way: suppose Alice knows $sk(k)$, then she can publish k so that anyone who wants to communicate with her can just encrypt by computing $F_k(m)$, which Alice can decode by inverting using $sk(k)$. They also suggested that the reverse process can actully serve as an *authentication* mechanism where Alice authenticates with $sk(k)$ by computing $t = F^{-1}(sk(k), m)$, and anyone knowing k can verify the integrity (as well as identity authentication).

This marked the invention of public-key *encryption* and *digital signature*. However, they didn't know how to instantiate such a magic box F. Later they met Merkle, and got inspired by Merkle's idea of key exchange. They were lucky to receive a suggestion of a mathematical tool from a Stanford colleague, and they came up with the famous *Diffie-Hellman* key-exchange protocol. This concrete protocol and their ingenious conceptual introduction of public-key cryptography were finally compiled in this ground-breaking paper "New Directions in Cryptography" [DH76]. In sum, their main contributions are

- Introducing **public-key cryptography**, including the notion of public-key *encryption* and *digital signature*, and an approach based on a magic box *trapdoor permutations*.
- Proposing the **DH key-exchange protocol** based on a number-theoretical problem which achieves the exponential gap that Merkle conjectured, if one is willing to accept certain assumptions on the computational hardness of the number-theoretical problem.

Finding a proper candidate for the magic box had to wait for another year by another three pioneers Rivest, Shamir and Adelmen [RSA78], who introduced the famous RSA functions. We hence have concrete public-key cryptosystems since.

Discovery at GCHQ. Interesting enough, according to a declassified document at the British intelligence agency GCHQ in 1997[2], similar ideas were developed even before the discovery in the public research community. Apparently in late 1960's, James Ellis envisied basically the same "magic" box that Diffie-Hellman later proposed and thought about using it for a new type of encryption – public-key encryption. He didn't know how to implement such a box either. He kept assigning this problem to new recruits until in 1973/74, Clifford Cocks and Malcolm Williamson came up with a solution essentially the same as RSA.

2.1 Diffie-Hellman key-exchange: abstract version

We discuss an abstract version of the DHKE protocol this time. Then we review some basic number theory in order to instantiate the DHKE protocol and also prepare ourselves for the following lectures.

Imagine two functions P, Q satisfying

1. Computing $P(x)$ is easy.
2. Computing $(x, P(y)) \mapsto Q(x,y)$ is easy.
3. Computing $(P(x), y) \mapsto Q(x,y)$ is easy.
4. Computing $(P(x), P(y)) \mapsto Q(x,y)$ is **hard**.

This enables the following key-exchange protocol

FS NOTE: Draw DHKE diagram

1. Alice picks random x, send $P(x)$ to Bob.
2. Similarly, Bob picks random x, send $P(x)$ to Bob.
3. Alice computes $Q(x,y)$ from $x, P(y)$, and Bob computes $Q(x,y)$ from $y, P(x)$.
4. Keep $Q(x,y)$ as their shared secret key.

Intuitively, any eavesdropper only sees $P(x)$ and $P(y)$ from which s/he cannot derive $Q(x,y)$ by our assumption.

3 Review: basic algebra and number theory

- Divisibility
- Prime numbers, prime numbers theorem, sample a random prime
- Modular arithmetic, greatest common divisor (GCD), (extended) Euclidean algorithm
- groups, Abelian group, $\mathbb{Z}_N^*$ and Z_p^* under modular multiplication, Euler's function

[2] Cf. http://cryptome.org/jya/ellisdoc.htm.

References

[DH76] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976. PDF at https://www-ee.stanford.edu/~hellman/publications/24.pdf.

[Mer78] Ralph C Merkle. Secure communications over insecure channels. *Communications of the ACM*, 21(4):294–299, 1978. PDF at http://www.merkle.com/1974/PuzzlesAsPublished.pdf.

[RSA78] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978. PDF at http://people.csail.mit.edu/rivest/Rsapaper.pdf.