The key-distribution problem
A public-key solution

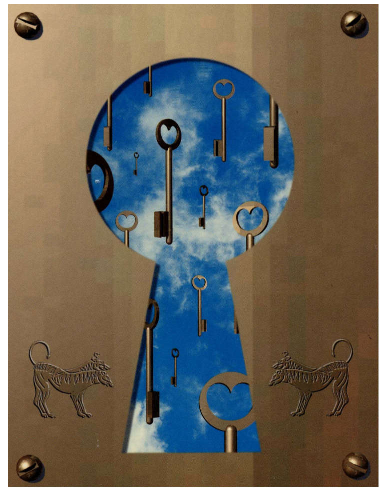
Foundations of Cryptography
Computer Science Department
Wellesley College

Fall 2016

Table of contents

The key-distribution problem

- Private-key cryptography requires shared, secret keys between each pair of communicating parties.
- How are all these keys shared in the first place?
- In situations where a large number of parties must pairwise, secretly communicate, many schemes do not scale well.



Key storage and secrecy

- When there are U employees, the number of secret keys is $\binom{U}{2} = \Theta(U^2)$ and every employee holds $U - 1$ keys.
- The situation is worse when employees must communicate with remote databases, servers, and so forth.
- All these keys need must be securely store.

Open systems

- Private-key cryptography can be used to solve the problem of secure communication in "closed" systems where it is possible to distribute secret keys via physical means.
- What happens when parties cannot physically meet, or where parties have transient interactions?

Key distribution centers (KDC)

All employees share a key with the KDC.

1. When Alice wants to communicate with Bob, she encrypts, using the secret key she shares with KDC: '`Alice wishes to communicate with Bob`'
2. The KDC chooses a new random key, called the *session key* and sends this to Alice (encrypted using Alice's shared key) and Bob (encrypted using Bob's shared key).
3. Alice and Bob communicate using the session key and destroy it when they are done.

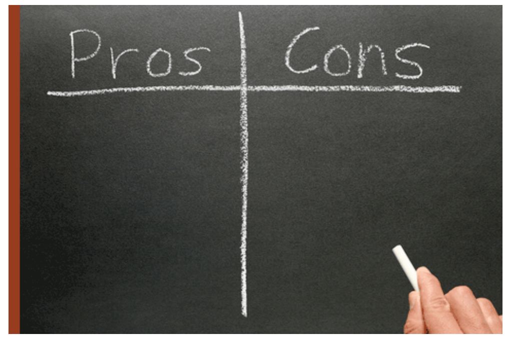
Good news/Bad news

Plus side:

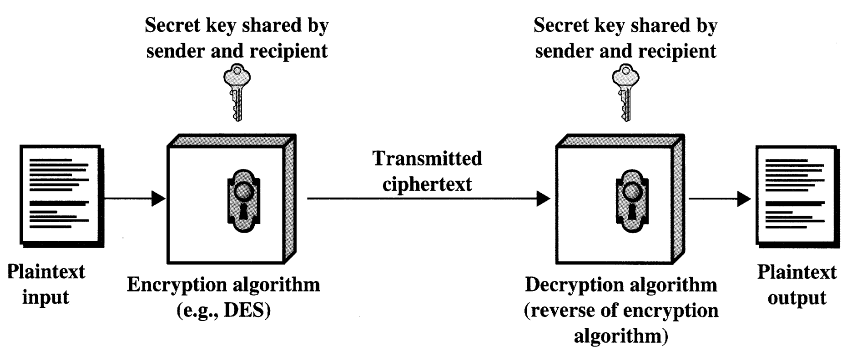
1. Each employee needs to store only *one* secret key. Limited storage devices, such as smart cards, could be used.
2. When an employee joins the organization all that must be done is set up a secret-key with the KDC. No other employees need be updated.

Minus side:

1. A successful attack on the KDC results in a complete break of security for all parties.
2. When the KDC is down, secure communications come to a halt.

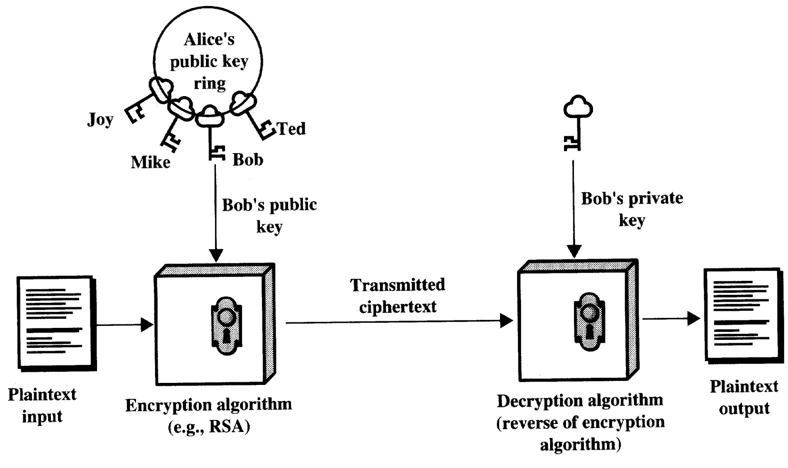


The state of affairs before 1976



After 1976, a new kid on the block

In 1976, Whitfield Diffie and Martin Hellman published a paper titled "New Directions in Cryptography" in which they proposed a completely new cryptographic paradigm.



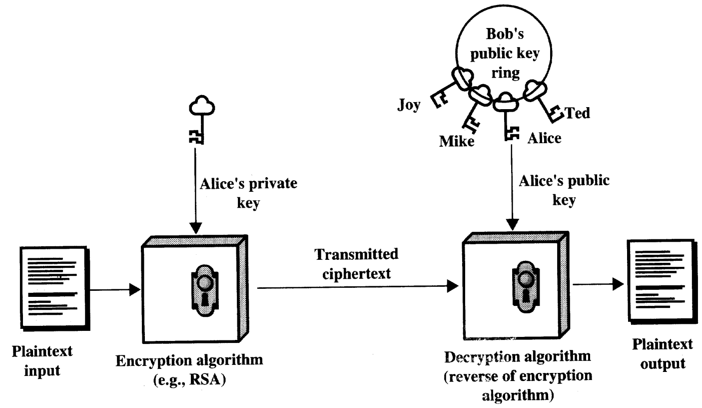
Addressing the limitations of private-key encryption*

1. Public-key allows key distribution to be done over public channels. Initial deployment and system maintenance is simplified.
2. Public-key vastly reduces the need to store many different secret keys. Even if a large number of pairs want to communicate secretly, each party needs store only one key: *her own*.
3. Finally, public-key is suitable for open environments where parties who have never previously interacted can communicate secretly.

*There are a fair number of details glossed over here, e.g., ensuring *authentic* distribution of public keys in the first place.

Digital signatures

In addition to the public-key encryption, Diffie and Hellman introduced a public-key analogue to message authentication codes, call *digital signatures*.



*Not only does this scheme prevent undetected tampering of a message, authenticity can be verified by anyone knowing the public key of the sender. *Nonrepudiation*: Alice cannot deny her signature.

Public-key implementation

- Although Diffie and Hellman introduced public-key encryption and digital signatures, they did not provide an implementation of either.
- A year later, Ron Rivest, Adi Shamir, and Len Adleman proposed the *RSA problem* and presented the first public-key encryption and digital signature schemes.

Implements of war

- Diffie and Hellman (and others publishing in cryptography) were under threat of prosecution.
- Under the *International Traffic in Arms Regulations*, technical literature on cryptography was considered an implement of war.

Introduction Key-Distribution Diffie-Hellman Exchange

Interactive key exchange

- Finally, in their now famous paper, Diffie and Hellman provided an implementation of an *interactive key exchange*.
- An interactive key exchange protocol is a method whereby parties who do not share any secret information can generate a shared, secret key by communicating over a public channel.

The setting

Alice and Bob run some protocol Π in order to generate a shared secret.

- Beginning with a security parameter 1^n, Alice and Bob choose (independent) random coins and run protocol Π:
- At the end of the protocol, Alice and Bob output keys $k_A, k_B \in \{0,1\}^n$, respectively.
- The basic correctness requirement is that $k_A = k_B$ for all choices of random coins.*

*Thus, we can speak of *the* key $k = k_A = k_B$.

A definition of security

The key-exchange experiment $\mathsf{KE}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n)$*:*

1. Two parties holding 1^n execute protocol Π resulting in a transcript trans containing all the messages sent by the parties, and a *key* k that is output by each of the parties.
2. A random bit $b \leftarrow \{0,1\}$ is chosen. If $b = 0$ then choose $\hat{k} \leftarrow \{0,1\}^n$ uniformly at random, and if $b = 1$ set $\hat{k} := k$.
3. $\mathcal{A}$ is given trans and $\hat{k}$, and outputs a bit b'.
4. The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise.

Definition 10.1 A key-exchange protocol Π is *secure in the presence of an eavesdropper* if for every probabilistic polynomial-time adversary $\mathcal{A}$ there exists a negligible function negl such that

$$\Pr[\mathsf{KE}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \mathsf{negl}(n).$$

*The Diffie-Hellman key-exchange protocol**

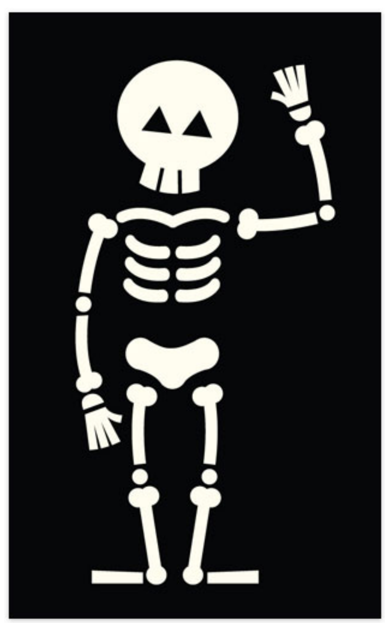
Construction 10.2.

- **Common input:** The security input 1^n
- **The protocol:**
  1. Alice runs $\mathcal{G}(1^n)$ to obtain $(\mathbb{G}, q, g)$.
  2. Alice chooses $x \leftarrow \mathbb{Z}_q$ uniformly at random, and computes $h_A := g^x$.
  3. Alice sends $(\mathbb{G}, q, g, h_A)$ to Bob.
  4. Bob receives $(\mathbb{G}, q, g, h_A)$. He chooses $y \leftarrow \mathbb{Z}_q$ uniformly at random and computes $h_B := g^y$. Bob sends h_B to Alice and outputs the key $k_B := h_A^y$.
  5. Alice receives h_B and outputs the key $k_A := h_B^x$.

*Checking correctness is easy.

Security of the Diffie-Hellman exchange

- At a bare bones minimum, in order for the Diffie-Hellman exchange to be secure it is necessary for the discrete logarithm problem to be hard relative to $\mathcal{G}$.
- However, this is not sufficient since is may be possible to compute the key $k_A = k_B$ without explicitly finding x or y.
- What is required is that g^{xy} be *indistinguishable from random* for any adversary given g, g^x, and g^y.



Decisional Diffie-Hellman (DDH) problem once more

The *decisional Diffie-Hellman (DDH) problem* is to distinguish $\mathrm{DH}_g(h_1, h_2)$ from a random group element for randomly chosen h_1, h_2.

Definition 8.63. We say that the *DDH problem is hard relative to $\mathcal{G}$* if for all probabilistic polynomial-time algorithms $\mathcal{A}$ there exists a negligible function negl such that

$$|\Pr[\mathcal{A}(\mathbb{G}, q, g, g^x, g^y, g^z) = 1] - \Pr[\mathcal{A}(\mathbb{G}, q, g, g^x, g^y, g^{xy}) = 1]| \leq \mathsf{negl}(n),$$

where in each case the probabilities are taken over the experiment in which $\mathcal{G}(1^n)$ outputs $(\mathbb{G}, q, g)$, and the random $x, y, z \in \mathbb{Z}_q$ are chosen.

Proof of security

Theorem 10.3. If the decisional Diffie-Hellman problem is hard relative to $\mathcal{G}$, then the Diffie-Hellman key-exchange protocol Π is secure in the presence of an eavesdropper (with respect to the experiment $\hat{\mathsf{KE}}^{\mathsf{eav}}_{\mathcal{A},\Pi}$.

Proof. Let $\mathcal{A}$ be a PPT adversary. Since $\Pr[b = 0] = \Pr[b = 1] = 1/2$, we have

$$\Pr\left[\hat{\mathsf{KE}}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n) = 1\right] = \frac{1}{2} \cdot \Pr\left[\hat{\mathsf{KE}}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n) = 1 \mid b = 1\right] + \frac{1}{2} \cdot \Pr\left[\hat{\mathsf{KE}}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n) = 1 \mid b = 0\right].$$

*Here $\hat{\mathsf{KE}}^{\mathsf{eav}}_{\mathcal{A},\Pi}$ stands for a modified experiment where if $b = 0$ the adversary is given $\hat{k} \leftarrow \mathbb{G}$ chosen uniformly at random.

The adversary's goal

In experiment $\hat{\mathsf{KE}}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n)$, adversary $\mathcal{A}$ receives $(\mathbb{G}, q, g, h_A, h_B, \hat{k})$, where $(\mathbb{G}, q, g, h_A, h_B)$ is the transcript of the protocol execution, and $\hat{k}$ is either the actual key g^{xy} (if $b = 1$) or a random group element (if $b = 0$).

Distinguishing between these two cases is exactly equivalent to solving the decisional Diffie-Hellman problem.*

*So are we really doing anything here?

Adversary's probability of success

$$
\begin{aligned}
&\Pr\left[\hat{\mathsf{KE}}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n) = 1\right] \\
&= \frac{1}{2} \cdot \Pr\left[\hat{\mathsf{KE}}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n) = 1 \mid b = 1\right] + \frac{1}{2} \cdot \Pr\left[\hat{\mathsf{KE}}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n) = 1 \mid b = 0\right] \\
&= \frac{1}{2} \cdot \Pr[\mathcal{A}(\mathbb{G}, g, q, g^x, g^y, g^{xy}) = 1] + \frac{1}{2} \cdot \Pr[\mathcal{A}(\mathbb{G}, g, q, g^x, g^y, g^z) = 0] \\
&= \frac{1}{2} \cdot \Pr[\mathcal{A}(\mathbb{G}, g, q, g^x, g^y, g^{xy}) = 1] + \frac{1}{2} \cdot (1 - \Pr[\mathcal{A}(\mathbb{G}, g, q, g^x, g^y, g^z) = 1]) \\
&= \frac{1}{2} + \frac{1}{2} \cdot (\Pr[\mathcal{A}(\mathbb{G}, g, q, g^x, g^y, g^{xy}) = 1] - \Pr[\mathcal{A}(\mathbb{G}, g, q, g^x, g^y, g^z) = 1]) \\
&\leq \frac{1}{2} + \frac{1}{2} \cdot |\Pr[\mathcal{A}(\mathbb{G}, g, q, g^x, g^y, g^{xy}) = 1] - \Pr[\mathcal{A}(\mathbb{G}, g, q, g^x, g^y, g^z) = 1]| \,.
\end{aligned}
$$

If the decisional Diffie-Hellman assumption is hard relative to $\mathcal{G}$, this the absolute value in the final line is bounded by some negligible runction negl, and

$$\Pr\left[\hat{\mathsf{KE}}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n) = 1\right] \leq \frac{1}{2} + \frac{1}{2} \cdot \mathsf{negl}(n).$$

$\square$