CSC3631 Cryptography
- Key Exchange

Thomas Gross

Quick Review I: $\mathbf{Z}_N$

What is $\mathbf{Z}_N$?

- The numbers {1, N, 2N, 3N, …}
- The numbers {1, 2, 3, … N}
- The numbers {0, 1, 2,… N-1}

Quick Review II: gcd

Which statements about the gcd() are false?

- **gcd**(a, b) = $s \cdot t + a \cdot b$ for some s and t
- **gcd**(a, b) = 1 if a and b are **coprime**.
- The running time of computing the **gcd**() with the Euclidian algorithm is **O**(a + b).

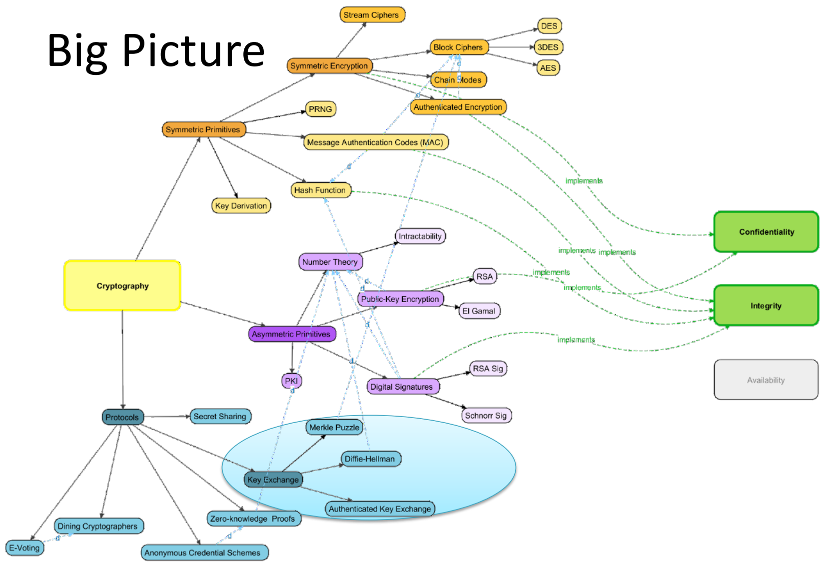
Quick Review III: Inverse

How do we find the multiplicative inverse in $\mathbf{Z}_N$?

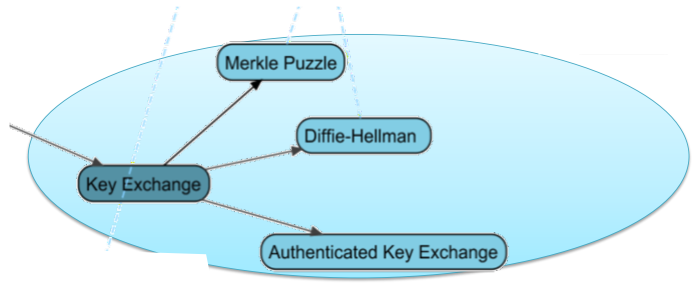
To compute the inverse of x:

- Compute the d = **gcd**(x, N); d is the inverse
- Compute $x^{-1} = (x \cdot N) - 1$
- Compute the Extended Euclidian algorithm for **gcd**(x, N) obtaining (d, s, t) with $d = s{\cdot}x + t{\cdot}N$; s is the inverse.

Big Picture



Big Picture

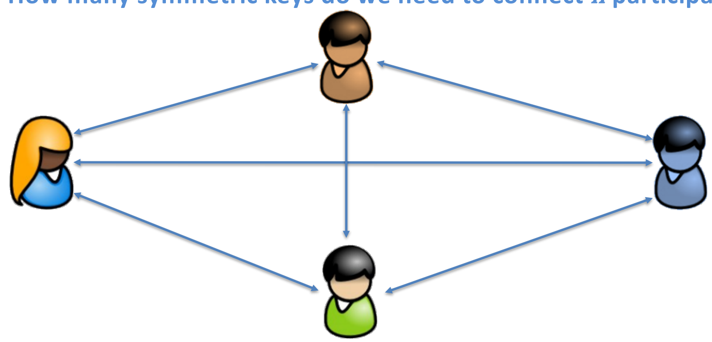


Roadmap

1. The Key Management Problem
2. Key Exchange
3. Symmetric Approach: Merkle Puzzles
4. Asymmetric Approach: Diffie-Hellman
5. Establishing Secure Channels

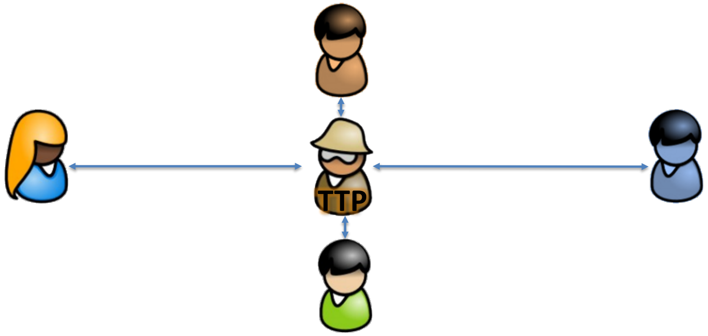
Key Management Problem

How many symmetric keys do we need to connect n participants?



Total Number of keys: O(n^2)
Safe storage of secret keys is difficult

Key Establishment w/ TTP



Number of key with a Trusted Third Party: **O**(n)

Trusted Third Party

Remember: Trusted Third Parties can be a great security risk!

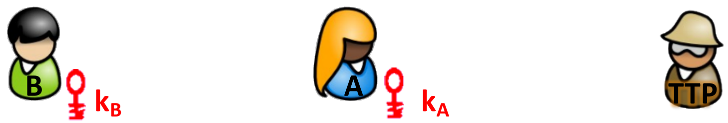
- A Trusted Third Party is a third party that can break your security policy

  --- Ross Anderson

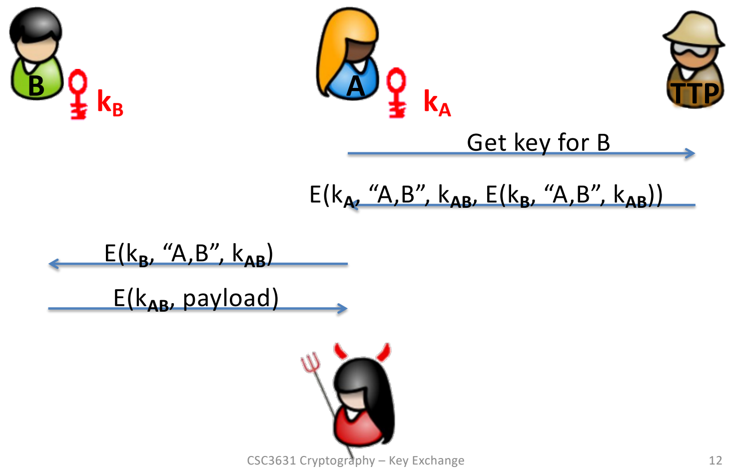
- In reality, a totally trustworthy third party doesn't exist.

Key Exchange with TTP

Only secure against eavesdropping.



How to attack this protocol?

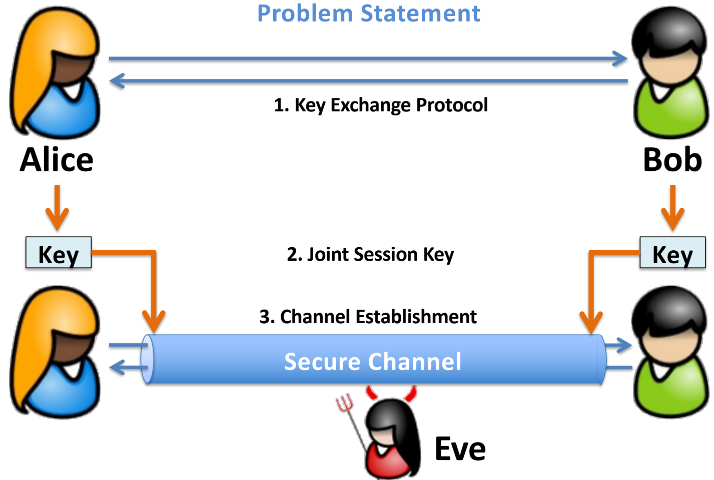


Roadmap

1. The Key Management Problem
2. **Key Exchange**
3. Symmetric Approach: Merkle Puzzles
4. Asymmetric Approach: Diffie-Hellman
5. Establishing Secure Channels

Key Exchange w/o TTP

Problem Statement



Definition of Key Exchange

- Consider two parties Alice and Bob.
- They run a *probabilistic protocol* (a set of instructions) on independent random bits.

- **Input:** security parameter 1^n.
- **Output:** Session keys k_A and k_B from $\{0, 1\}^n$.

- **Correctness:** $k_A = k_B = K$
  (in an honest execution of the protocol)

Security of Key Exchanges

A Game with the Adversary

Given: The parties with security parameter 1^n execute a key exchange protocol and output a transcript ***trans*** and a key K.

1. Flip a fair coin; obtain a uniformly random bit b in {0,1}
   - If $b = 0$ then set $K^* := K$
   - If $b = 1$ then choose K^* in $\{0, 1\}^n$ uniformly random.
2. **Input** for Adversary **A**: ***trans***, K^*
3. **Output** of Adversary **A**: a bit b'

Adversary A success: if $b' = b$.

Security of Key Exchanges

Informally stated, a key exchange protocol is secure if any adversary can only do negligibly better than throwing a coin to choose *b'*.

Key Exchange Security:
A key exchange protocol is secure in presence of an eavesdropper, if for all probabilistic polynomial-time* adversaries A their success probability is only negligibly* better than ½.

[*) Lecture 14 will treat polynomial-time and negligible in more detail.]

Observations on the Security of Key Exchange

- A key exchange is secure, if the session key *K* output by Alice and Bob is *completely* ***unguessable*** by an adversary.
- This is defined formally by requiring that an eavesdropping adversary should be unable to ***distinguish*** an actual session key *K* (generated by the protocol) from a uniform random bitstring of length *n* (independent of transcript **trans**).
- Indistinguishability is much stronger than requiring that the adversary be unable to *compute* the session key *K.*

Quick Review

- Static symmetric keys are impractical
- Trusted Third Parties bear a security risk.
- We need ad-hoc key exchange to establish a joint **session key**.
- **Main goal:** establish a secure channel

Roadmap

1. The Key Management Problem
2. Key Exchange
3. **Symmetric Approach: Merkle Puzzles**
4. Asymmetric Approach: Diffie-Hellman
5. Establishing Secure Channels

Merkle Puzzle Key Exchange

How can we solve key exchange symmetrically?

Only secure against eavesdropping.
Only use symmetric primitives.

Idea: Create a Puzzle, Difficult to Solve

Take a function difficult to solve as basis

- Decrypting AES by brute-force key search

Tailor the difficulty such that it can be solved

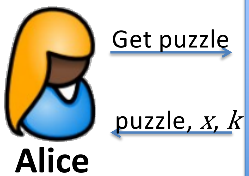
- Choose AES key with leading zeros:

$$K \leftarrow \{0\}^{128-l}\,\{0,1\}^{l}$$

Needs 2^l brute force decryption attempts to solve

Puzzle Generator

How to create a single puzzle?



Puzzle Generator PGen

Choose random P **in** $\{0,1\}^{32}$

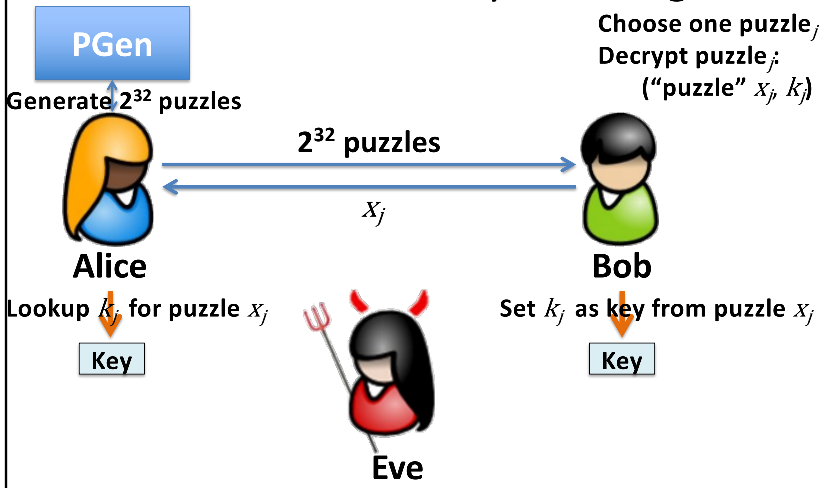
Choose random x, k **in** $\{0, 1\}^{128}$

puzzle $\leftarrow$ **E**$((\{0\}^{96}||P)$, "puzzle"x, $k)$

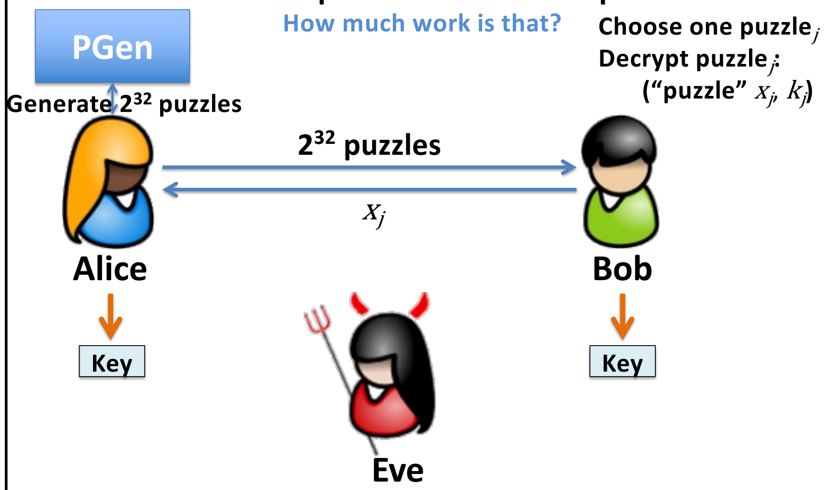
Remember the puzzle structure:

$$\mathbf{E}(\underbrace{(\{0\}^{96}||P)}_{\text{Random encryption key}},\ \underbrace{\text{“puzzle”}||x}_{\text{Random ID}},\ \underbrace{k}_{\text{Random session key}})$$

Merkle-Puzzle Key Exchange



Computational Gap



Quick Review

- **Idea:** Many puzzles to choose from
- Brute-force decryption of block cipher

- Workload Alice and Bob: **O**(n), e.g., 2^{32}
- Workload for Eve: **O**(n^2), e.g., 2^{64}

- Quadratic gap best known for black-box symmetric block ciphers.

Roadmap

1. The Key Management Problem
2. Key Exchange
3. Symmetric Approach: Merkle Puzzles
4. **Asymmetric Approach: Diffie-Hellman**
5. Establishing Secure Channels

Diffie-Hellman Key Exchange

Only secure against eavesdropping.
Use **asymmetric** primitives.

The Structure of $(\mathbf{Z}_p)^*$

Recall: What's our environment for key exchange?

The set of invertible elements of $\mathbf{Z}_p$ is **cyclic**.

$(\mathbf{Z}_p)^*$ is a **cyclic group**

Exists a g **in** $(\mathbf{Z}_p)^*$ such that

$\{ 1, g, g^2, g^3, \ldots g^{p-2} \} \quad = \quad (\mathbf{Z}_p)^*$

We call g a **generator** of $(\mathbf{Z}_p)^*$

Example p=5: $\{1, 3, 3^2, 3^3\} = \{1, 3, 4, 2\} = (\mathbf{Z}_5)^*$

Idea: Difficult Algebraic Function

Give adversary Eve

- Group definition: p, generator g

  (creating prime-order subgroup G of $(\mathbf{Z}_p)^*$)
- Two elements: g^a, g^b

(Computational) Diffie-Hellman Problem is said intractable:

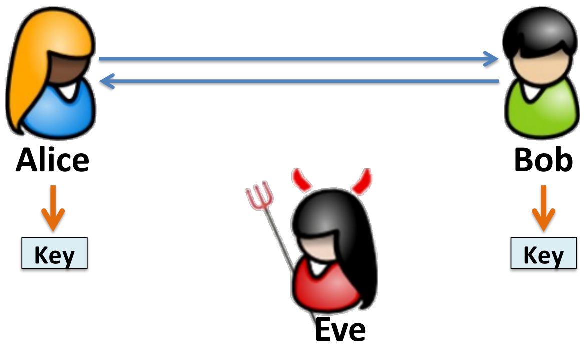
$\mathbf{DH}_g(g^a, g^b) = g^{ab}$ in $(\mathbf{Z}_p)^*$

Diffie-Hellman in $(\mathbf{Z}_p)^*$

How to create a key exchange from the DH problem?

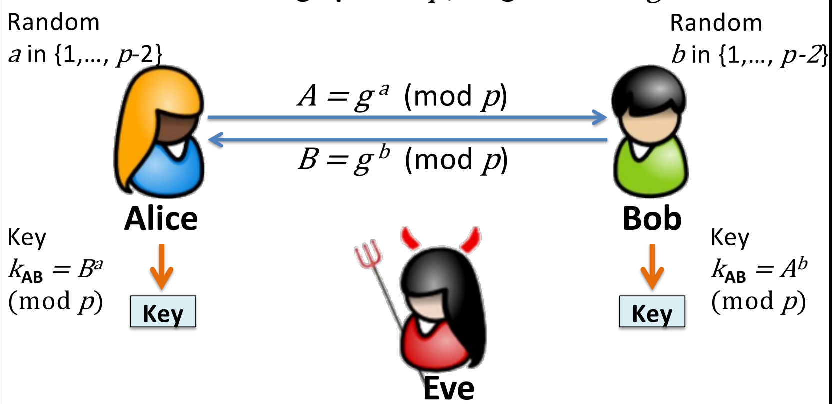
Given: large prime p; **generator** g,

generating prime-order (q) sub-group G of $(\mathbf{Z}_p)^*$



Diffie-Hellman in $(\mathbf{Z}_p)^*$

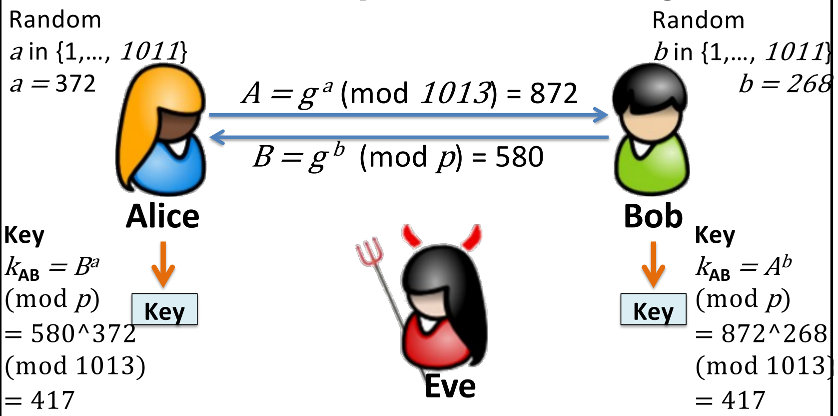
Given: large prime p; **generator** g



[Note: Full DH setup is in a sub-group G of $(\mathbf{Z}_p)^*$ with prime order q]

Example Diffie-Hellman in $(\mathbf{Z}_{1013})^*$

Given: large prime $p=1013$; **generator** $g=3$



How hard is Diffie-Hellman?

According to RSA Security:

Symmetric Cipher Key Size	DH/RSA Modulus Size
80 bits	**1024 bits**
128 bits	**3072 bits**

Best known algorithm to break DH/RSA:

General Number Field Sieve

Expected running time **O**(**exp**(**log**(n)$^{1/3}$)

"Are 1024-bit RSA keys are dead?" Arien Lenstra: "The answer to that question is an unqualified yes."

[See Shoup2008, Section 15.5]

Quick Review

- Diffie-Hellman is an asymmetric key exchange
- Based on the hardness of computing and distinguishing
  $\mathbf{DH}_g(g^a, g^b) = g^{ab}$ in $(\mathbf{Z}_p)^*$
- One needs much larger keys for asymmetric algorithms to be secure, than for symmetric ones.
- **Use at least 3072-bit keys for DH/RSA.**

Roadmap

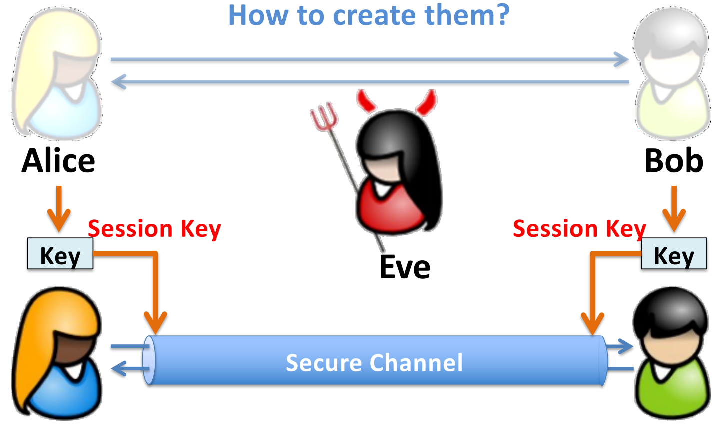
1. The Key Management Problem
2. Key Exchange
3. Symmetric Approach: Merkle Puzzles
4. Asymmetric Approach: Diffie-Hellman
5. **Establishing Secure Channels**

Quiz

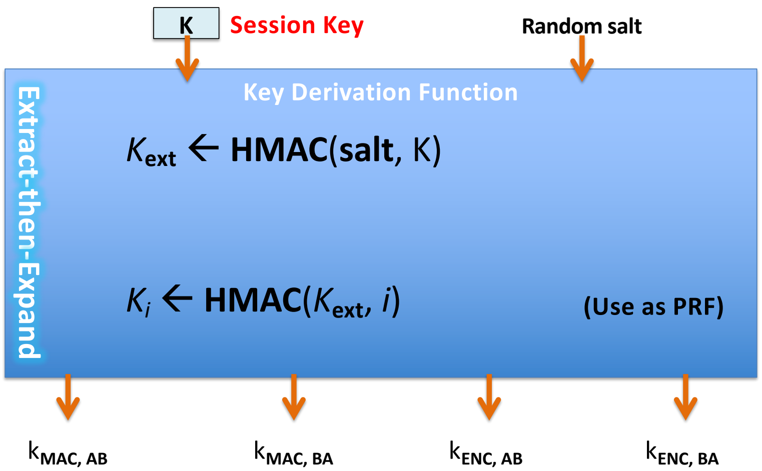
- What are secure channel examples used in daily life?
- What are secure channels good for?

Secure Channels

How to create them?



Key Derivation and Channel Creation



Quick Review

- Single **session key** as input from key exchange.
- Need to get **keys for communication**:
  - Symmetric cipher (confidentiality)
  - Message Authentication Code (Integrity)
- **Key Derivation Function (KDF)**
- Generate one separate key per use and direction.