When the math fails
Side-channel attacks on ECDH

Nick Egbert

Student Colloquium Talk

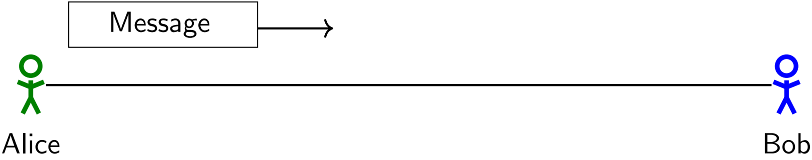
2 October 2019

Overview

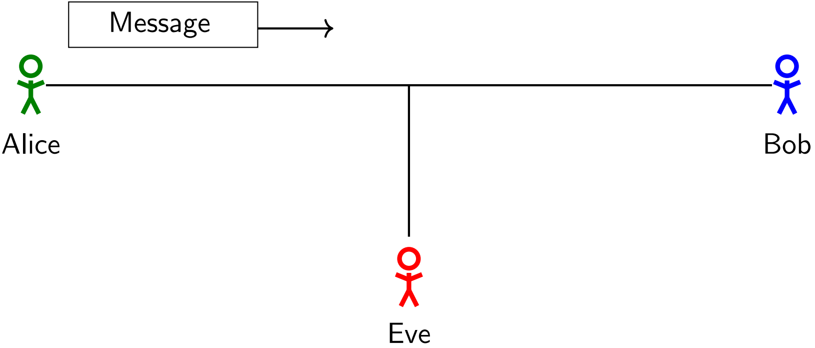
1. Cryptography basics
   - The general problem
   - Classical Diffie-Hellman
2. Elliptic curve basics
   - What they look like
   - Group structure
   - Montgomery curves
3. ECDH
4. Side channel attacks
5. Bonus

The problem

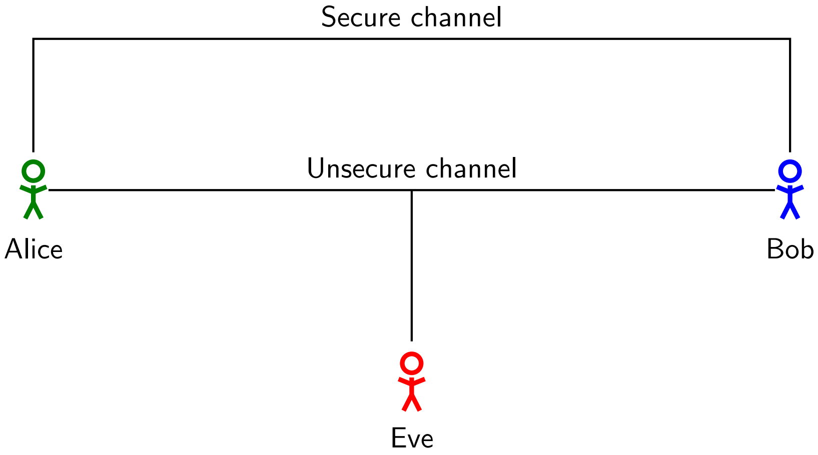
The problem



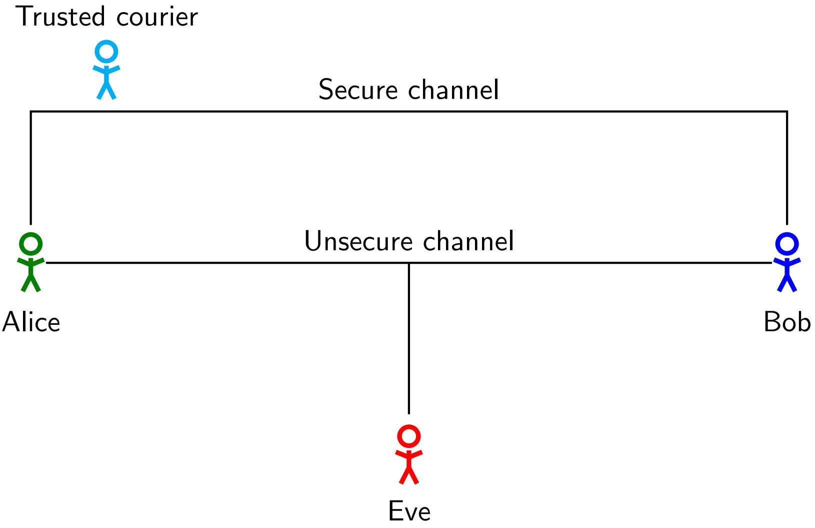
The problem



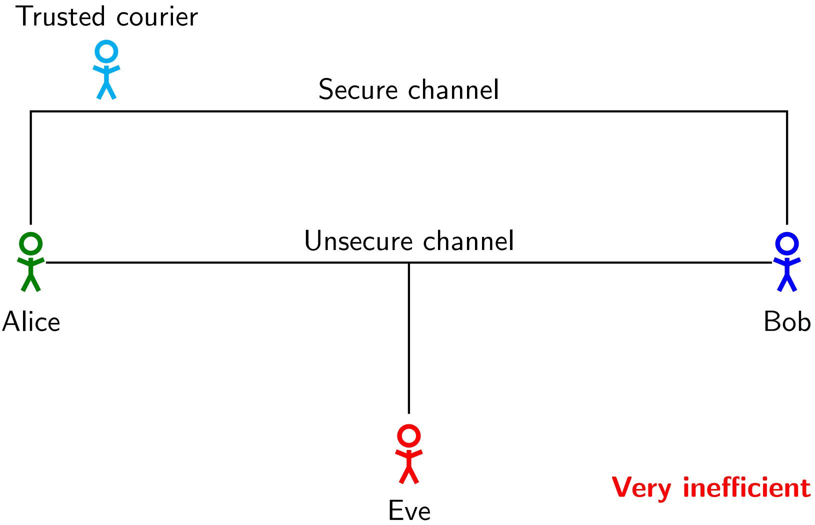
The problem



The problem


Trusted courier
Secure channel
Unsecure channel
Alice
Bob
Eve

The problem



Very inefficient

The solution

- There are two basic types of encryption: symmetric and asymmetric.

The solution

- There are two basic types of encryption: symmetric and asymmetric.
- In symmetric encryption, both parties have the same key for encrypting and decrypting.

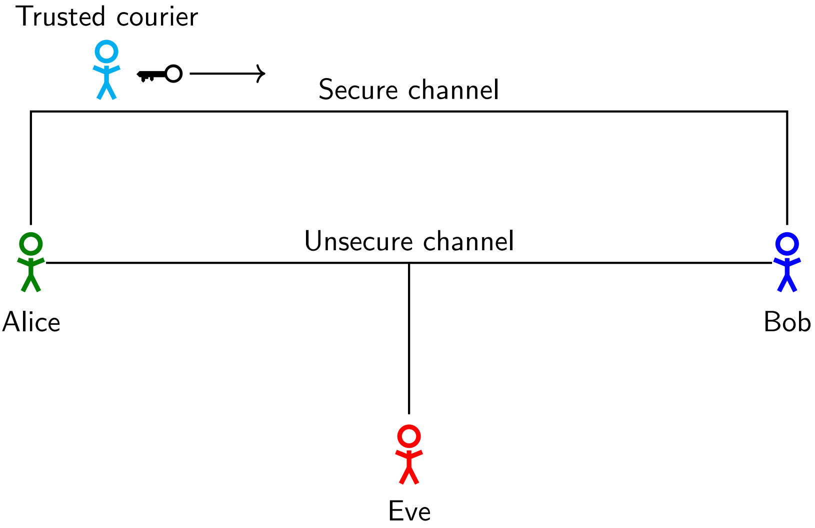
The solution

- There are two basic types of encryption: symmetric and asymmetric.
- In symmetric encryption, both parties have the same key for encrypting and decrypting.
- Asymmetric encryption is not symmetric.

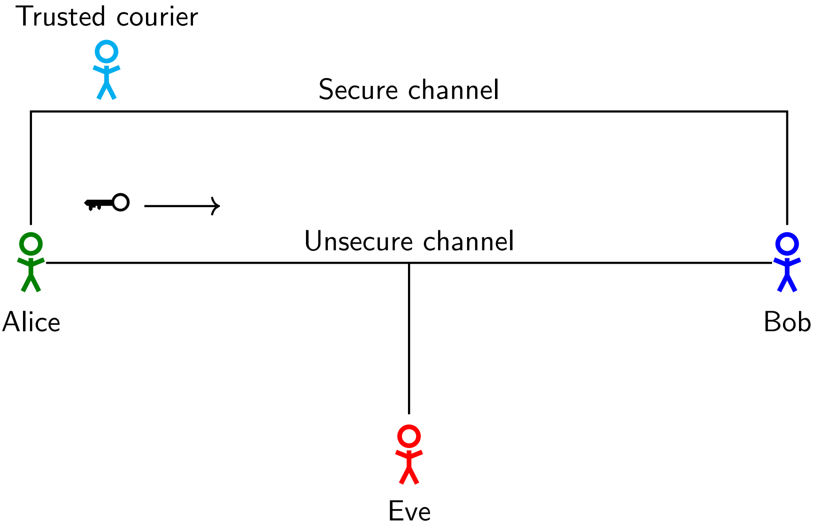
The solution

- There are two basic types of encryption: symmetric and asymmetric.
- In symmetric encryption, both parties have the same key for encrypting and decrypting.
- Asymmetric encryption is not symmetric.
- Asymmetric encryption is generally used to establish a shared key.

The solution



The solution



Discrete log problem (DLP)

Let p be a prime number, and let $a, b \in \mathbb{Z}$ such that $a, b \not\equiv 0 \bmod p$. Suppose we know there exists $k \in \mathbb{Z}$ such that

$$a^k \equiv b \pmod{p}.$$

The **(classical) discrete log problem** is to find k.

Discrete log problem (DLP)

Let p be a prime number, and let $a, b \in \mathbb{Z}$ such that $a, b \not\equiv 0 \bmod p$.
Suppose we know there exists $k \in \mathbb{Z}$ such that

$$a^k \equiv b \pmod{p}.$$

The **(classical) discrete log problem** is to find k.
More generally, if G is a group and $a, b \in G$, and given

$$a^k = b,$$

the discrete log problem is to find k.

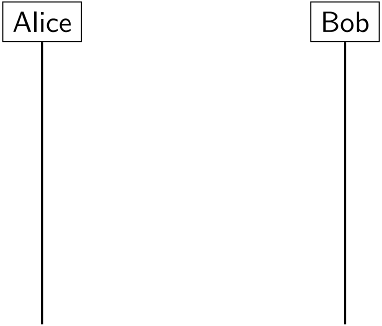
Diffie-Hellman Key Exchange (1976)

Public parameters:
g, p

Diffie-Hellman Key Exchange (1976)

Public parameters:

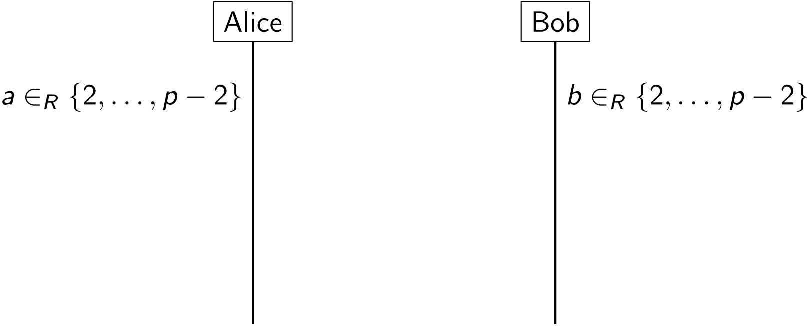
g, p



Diffie-Hellman Key Exchange (1976)

Public parameters:

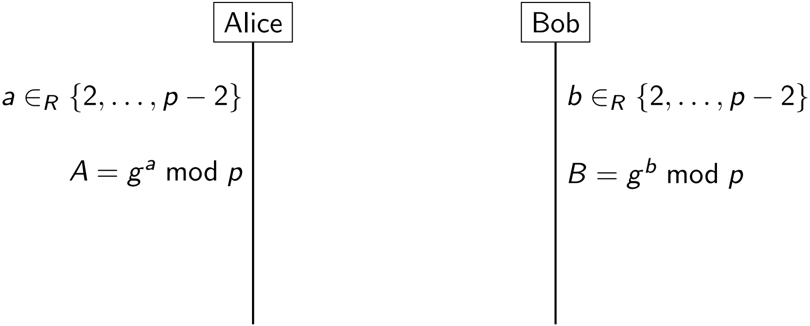
g, p



Diffie-Hellman Key Exchange (1976)

Public parameters:

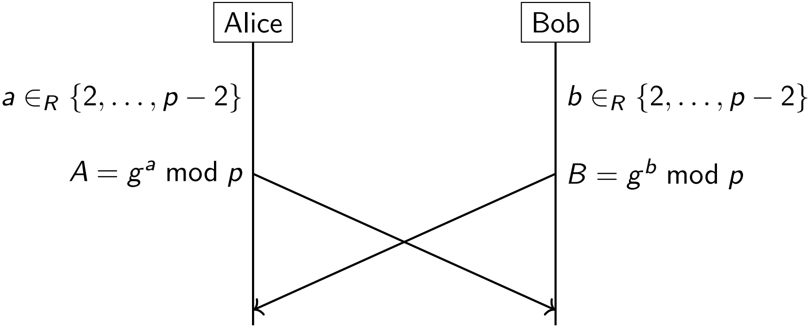
g, p



Diffie-Hellman Key Exchange (1976)

Public parameters:

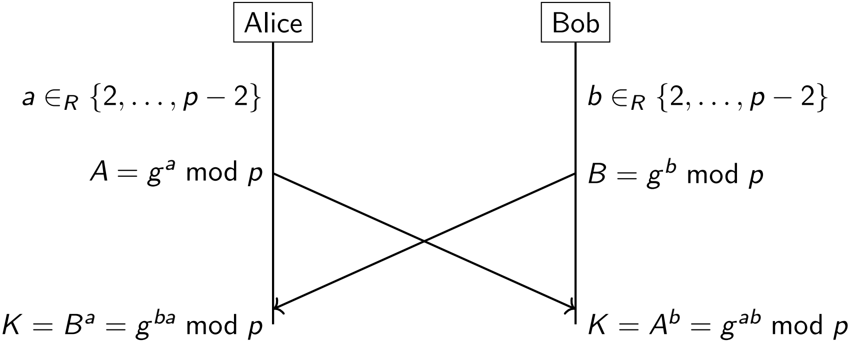
g, p



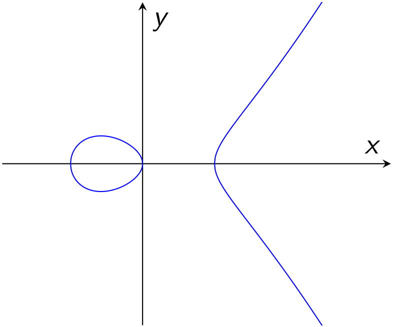
Diffie-Hellman Key Exchange (1976)

Public parameters:

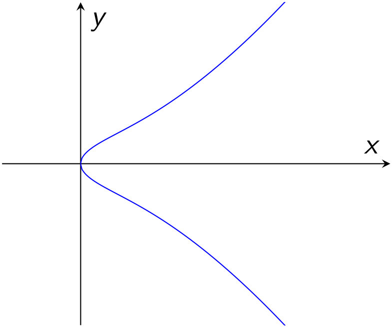
g, p



Elliptic curves

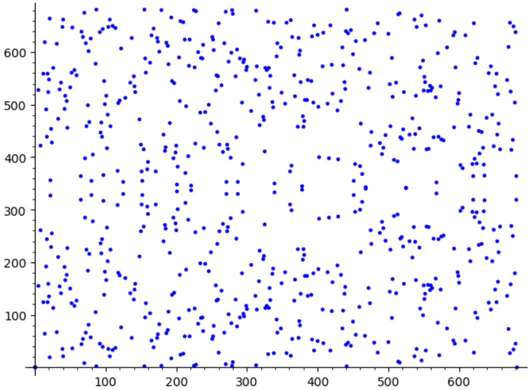


$$E_1 \colon y^2 = x^3 - x$$



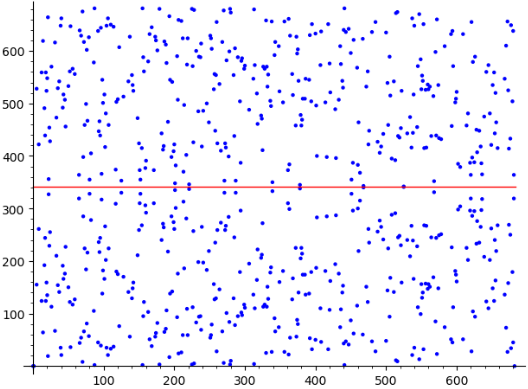
$$E_2 \colon y^2 = x^3 + x$$

Elliptic curve modulo p



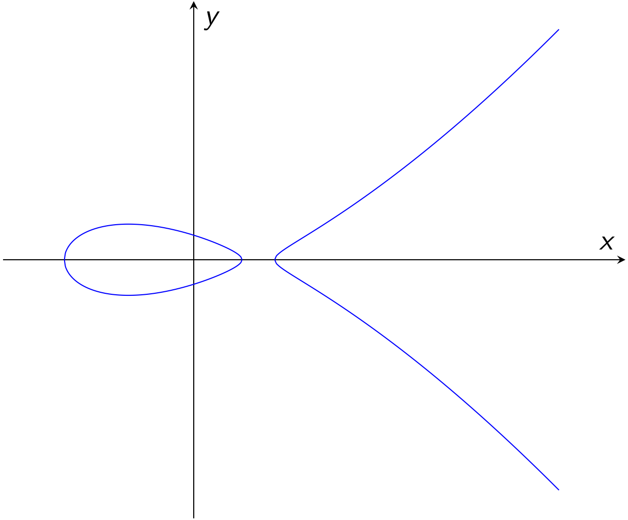
$$E_1 : y^2 = x^3 - x \pmod{683}$$

Elliptic curve modulo p

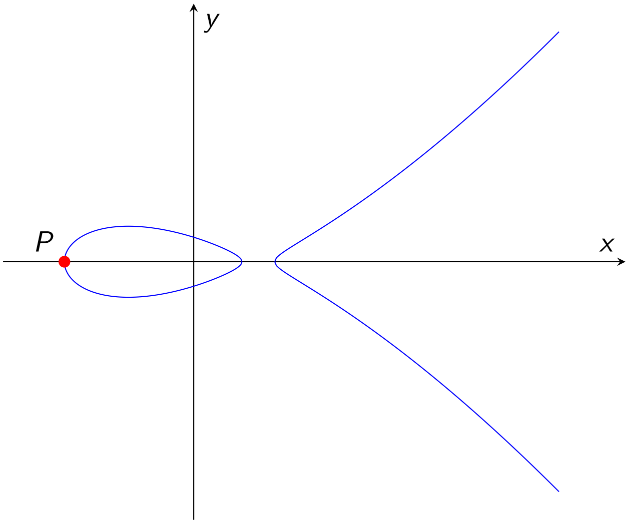


$$E_1 : y^2 = x^3 - x \pmod{683}$$

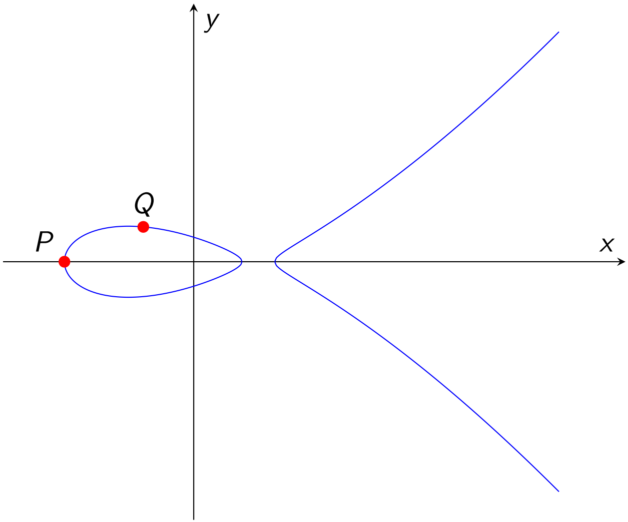
Group structure



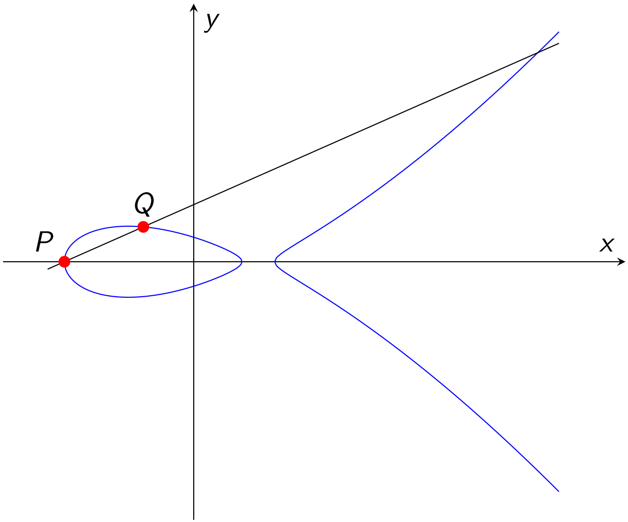
Group structure



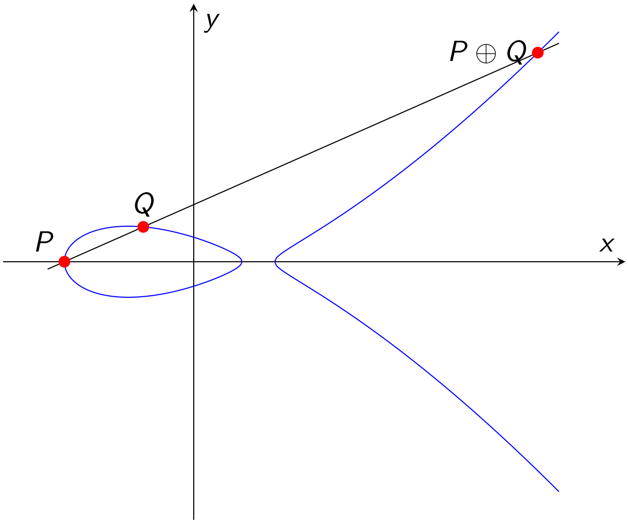
Group structure



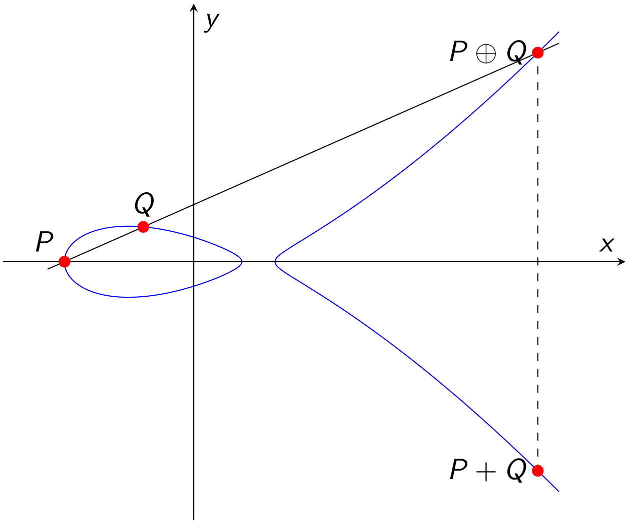
Group structure



Group structure



Group structure



Elliptic curves

- For crypto, we work in $\mathbb{F}_q$, where $q = p^n$ is a prime power. We assume $p \neq 2, 3$.
- An elliptic curve $E/\mathbb{F}_q$ is a nonsingular curve satisfying the cubic equation
$$y^2 = x^3 + Ax + B.$$
- The set of points on E lying in $\mathbb{F}_q$ plus the point at infinity turns E into a group, denoted $E(\mathbb{F}_q)$.
- Points are added using chord-and-tangent method

Elliptic curves

- For crypto, we work in $\mathbb{F}_q$, where $q = p^n$ is a prime power. We assume $p \neq 2, 3$.
- An elliptic curve $E/\mathbb{F}_q$ is a nonsingular curve satisfying the cubic equation
$$y^2 = x^3 + Ax + B.$$
- The set of points on E lying in $\mathbb{F}_q$ plus the point at infinity turns E into a group, denoted $E(\mathbb{F}_q)$.
- Points are added using chord-and-tangent method

Elliptic curves

- For crypto, we work in $\mathbb{F}_q$, where $q = p^n$ is a prime power. We assume $p \neq 2, 3$.
- An elliptic curve $E/\mathbb{F}_q$ is a nonsingular curve satisfying the cubic equation

$$y^2 = x^3 + Ax + B.$$

- The set of points on E lying in $\mathbb{F}_q$ plus the point at infinity turns E into a group, denoted $E(\mathbb{F}_q)$.
- Points are added using chord-and-tangent method

Elliptic curves

- For crypto, we work in $\mathbb{F}_q$, where $q = p^n$ is a prime power. We assume $p \neq 2, 3$.
- An elliptic curve $E/\mathbb{F}_q$ is a nonsingular curve satisfying the cubic equation
$$y^2 = x^3 + Ax + B.$$
- The set of points on E lying in $\mathbb{F}_q$ plus the point at infinity turns E into a group, denoted $E(\mathbb{F}_q)$.
- Points are added using chord-and-tangent method

Montgomery curves

- For efficiency reasons, Montgomery curves are used in practice.
- A Montgomery curve is a nonsingular E cubic curve satisfying

$$By^2 = x^3 + Ax^2 + x.$$

- All Montgomery curves are elliptic curves, but not conversely.
- The j-invariant of a Montgomery curve E is given by

$$j(E_{A,B}) = \frac{256(A^2 - 3)^3}{A^2 - 4}$$

Montgomery curves

- For efficiency reasons, Montgomery curves are used in practice.
- A Montgomery curve is a nonsingular E cubic curve satisfying

$$By^2 = x^3 + Ax^2 + x.$$

- All Montgomery curves are elliptic curves, but not conversely.
- The j-invariant of a Montgomery curve E is given by

$$j(E_{A,B}) = \frac{256(A^2 - 3)^3}{A^2 - 4}$$

Montgomery curves

- For efficiency reasons, Montgomery curves are used in practice.
- A Montgomery curve is a nonsingular E cubic curve satisfying

$$By^2 = x^3 + Ax^2 + x.$$

- All Montgomery curves are elliptic curves, but not conversely.
- The j-invariant of a Montgomery curve E is given by

$$j(E_{A,B}) = \frac{256(A^2 - 3)^3}{A^2 - 4}$$

Montgomery curves

- For efficiency reasons, Montgomery curves are used in practice.
- A Montgomery curve is a nonsingular E cubic curve satisfying

$$By^2 = x^3 + Ax^2 + x.$$

- All Montgomery curves are elliptic curves, but not conversely.
- The j-invariant of a Montgomery curve E is given by

$$j(E_{A,B}) = \frac{256(A^2 - 3)^3}{A^2 - 4}$$

ECDH

- ECDH = "Elliptic Curve Diffie-Hellman"
- Alice and Bob agree on an elliptic curve E and a field $\mathbb{F}_q$ such that the DLP is hard for $E(\mathbb{F}_q)$.
- We require that $q + 1 - \#E(\mathbb{F}_q) \neq 0 \mod p$.
- They agree on a point $P \in E(\mathbb{F}_q)$ of large (usually prime) order.

ECDH

- ECDH = "Elliptic Curve Diffie-Hellman"
- Alice and Bob agree on an elliptic curve E and a field $\mathbb{F}_q$ such that the DLP is hard for $E(\mathbb{F}_q)$.
- We require that $q + 1 - \#E(\mathbb{F}_q) \not\equiv 0 \mod p$.
- They agree on a point $P \in E(\mathbb{F}_q)$ of large (usually prime) order.

ECDH

- ECDH = "Elliptic Curve Diffie-Hellman"
- Alice and Bob agree on an elliptic curve E and a field $\mathbb{F}_q$ such that the DLP is hard for $E(\mathbb{F}_q)$.
- We require that $q + 1 - \#E(\mathbb{F}_q) \not\equiv 0 \mod p$.
- They agree on a point $P \in E(\mathbb{F}_q)$ of large (usually prime) order.

ECDH

- ECDH = "Elliptic Curve Diffie-Hellman"
- Alice and Bob agree on an elliptic curve E and a field $\mathbb{F}_q$ such that the DLP is hard for $E(\mathbb{F}_q)$.
- We require that $q + 1 - \#E(\mathbb{F}_q) \not\equiv 0 \mod p$.
- They agree on a point $P \in E(\mathbb{F}_q)$ of large (usually prime) order.

ECDH

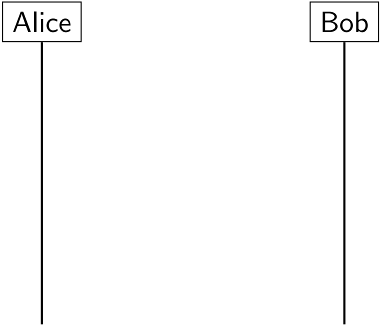
Public parameters:

$E(\mathbb{F}_q)$, P

ECDH

Public parameters:

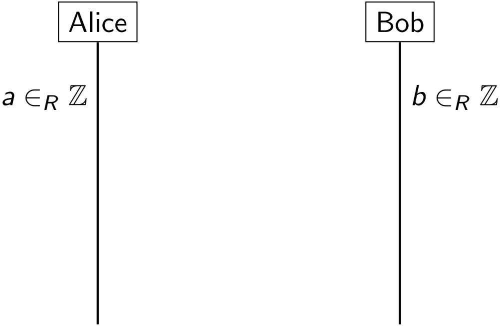
$E(\mathbb{F}_q)$, P



ECDH

Public parameters:

$E(\mathbb{F}_q)$, P



ECDH

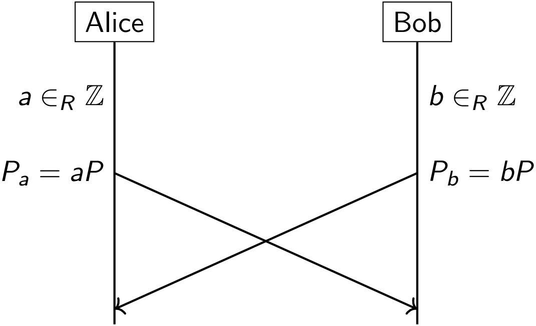
Public parameters:
$E(\mathbb{F}_q)$, P

Alice	Bob
$a \in_R \mathbb{Z}$	$b \in_R \mathbb{Z}$
$P_a = aP$	$P_b = bP$

ECDH

Public parameters:

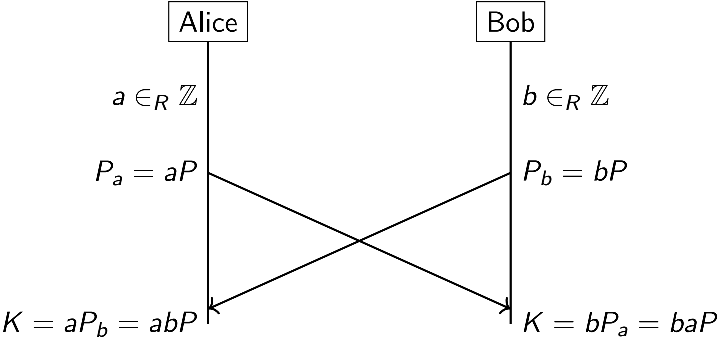
$E(\mathbb{F}_q)$, P



ECDH

Public parameters:

$E(\mathbb{F}_q)$, P



EC DLP

Let P, Q be points on $E(\mathbb{F}_q)$. Suppose we know there exists $d \in \mathbb{Z}$ such that

$$Q = dP.$$

The **elliptic curve discrete log problem** is to find d.

Advantages of ECDH

- Using elliptic curves allows for *much* smaller key sizes: an RSA 4096-bit key provides the same level of security as a 313-bit EC key.
- The group law for elliptic curves can be performed efficiently.
- No known subexponential algorithm to solve DLP in this setting.

Advantages of ECDH

- Using elliptic curves allows for *much* smaller key sizes: an RSA 4096-bit key provides the same level of security as a 313-bit EC key.
- The group law for elliptic curves can be performed efficiently.
- No known subexponential algorithm to solve DLP in this setting.

Advantages of ECDH

- Using elliptic curves allows for *much* smaller key sizes: an RSA 4096-bit key provides the same level of security as a 313-bit EC key.
- The group law for elliptic curves can be performed efficiently.
- No known subexponential algorithm to solve DLP in this setting.

Side-channel analysis

Side-channel attack	An attack on the physical implementation of a cryptosystem
Timing attack	Watch CPU clock to measure how long it takes to perform various operations
Simple power analysis (SPA)	Monitor power consumption during cryptographic process
Differential power analysis (DPA)	Statistically analyze power consumption measurements from a cryptosystem

Scalar multiplication in E

Algorithm 1: double-and-add

Input: Binary integer $d = (d_{\ell-1}, \ldots, d_0)$ and a point $P \in E$.
Output: Point $Q = dP \in E$.

$Q \leftarrow P$
for i from $\ell - 2$ to 0 **do**
    $Q \leftarrow 2Q$
    if $d_i = 1$ **then**
        $Q \leftarrow Q + P$
return Q

Weaknesses in Algorithm 1

- Addition and doubling require different amounts of power and CPU time
- This makes Algorithm 1 vulnerable to timing attacks and SPA

SPA-resistant scalar multiplication in E

Algorithm 2: Montgomery ladder

Input: Binary integer $d = (d_{\ell-1}, \ldots, d_0)$ and a point $P \in E$.
Output: Point $R_0 = dP \in E$.

$R_0 \leftarrow P$
$R_1 \leftarrow 2P$
for i from $\ell - 2$ to 0 **do**
  if $d_i = 0$ **then**
    $R_0 \leftarrow R_0 + R_1$
    $R_1 \leftarrow 2R_1$
  else
    $R_1 \leftarrow R_0 + R_1$
    $R_0 \leftarrow 2R_0$
return R_0

Differential power analysis

- The jth step of Algorithm 2 depends on the bits $d_{\ell-1}, \ldots, d_j$ of d.

Differential power analysis

- The jth step of Algorithm 2 depends on the bits $d_{\ell-1}, \ldots, d_j$ of d.
- Running the algorithm several times reveals statistical correlations that can aid to recover the bits of d.

Differential power analysis

- The jth step of Algorithm 2 depends on the bits $d_{\ell-1}, \ldots, d_j$ of d.
- Running the algorithm several times reveals statistical correlations that can aid to recover the bits of d.
- Other countermeasures are needed.

Randomizing d

- We wish to compute $Q = dP$, keeping d secret.

Randomizing d

- We wish to compute $Q = dP$, keeping d secret.
- Write $d' = d + kN$, where k is a random integer and $N = \#E(\mathbb{F}_q)$.

Randomizing d

- We wish to compute $Q = dP$, keeping d secret.
- Write $d' = d + kN$, where k is a random integer and $N = \#E(\mathbb{F}_q)$.
- Compute $Q = d'P = dP$.

Blinding the point P

- Secretly store $S = dR$.

Blinding the point P

- Secretly store $S = dR$.
- Compute $dP' = d(R + P)$.

Blinding the point P

- Secretly store $S = dR$.
- Compute $dP' = d(R + P)$.
- Recover $dP = dP' - S$.

Blinding the point P

- Secretly store $S = dR$.
- Compute $dP' = d(R + P)$.
- Recover $dP = dP' - S$.
- On next iteration set $R \leftarrow (-1)^b 2R$ and $S \leftarrow (-1)^b 2S$ for a random bit b.

Randomized projective coordinates

- We projectivize E by

$$E\colon\ Y^2Z = X^3 + AX^2Z + BZ^3,$$

where $x = X/Z$, $y = Y/Z$.

Randomized projective coordinates

- We projectivize E by

$$E\colon Y^2Z = X^3 + AX^2Z + BZ^3,$$

  where $x = X/Z$, $y = Y/Z$.
- In projective space $(X : Y : Z)$ and $(\lambda X : \lambda Y : \lambda Z)$ are equivalent points on E for any scalar λ.

Randomized projective coordinates

- We projectivize E by

$$E\colon\ Y^2Z = X^3 + AX^2Z + BZ^3,$$

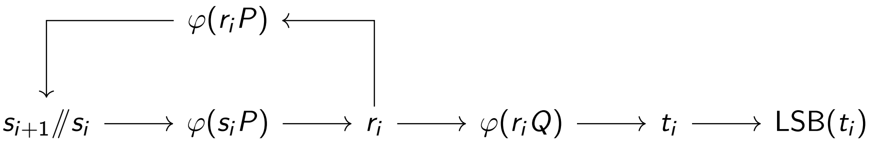
  where $x = X/Z$, $y = Y/Z$.
- In projective space $(X : Y : Z)$ and $(\lambda X : \lambda Y : \lambda Z)$ are equivalent points on E for any scalar λ.
- Given $P = (X : Y : Z)$, choose random integer λ and use $P' = (\lambda X : \lambda Y : \lambda Z)$.

Dual_EC_DRBG

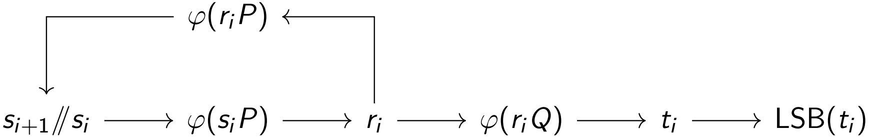
- $p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$
- $E\colon y^2 =$ $x^3 - 3x + 4105836372515214212932612978004726840911444101599\backslash\backslash$ $3725554835256314039467401291$
- P a generator of $E(\mathbb{F}_p)$
- $Q \in E(\mathbb{F}_p)$ a specified constant

How it works

- $\varphi\colon E(\mathbb{F}_p) \to \mathbb{Z}$ via $\varphi(x, y) = x$
- s_0 a randomly chosen seed
- $r_i = \varphi(s_i P)$, $t_i = \varphi(r_i Q)$, $s_{i+1} = \varphi(r_i P)$

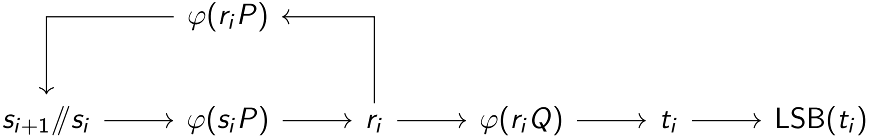


The attack



- Goal: Determine s_i for some i.

The attack



- Goal: Determine s_i for some i.
- $\text{LSB}(t_i)$ only cuts off 16 bits of information.

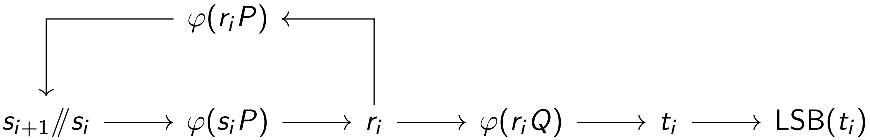
The attack

$$r_i \longrightarrow \varphi(r_i P) \longrightarrow s_{i+1}$$

$$s_{i+1} /\!/ s_i \longrightarrow \varphi(s_i P) \longrightarrow r_i \longrightarrow \varphi(r_i Q) \longrightarrow t_i \longrightarrow \mathrm{LSB}(t_i)$$

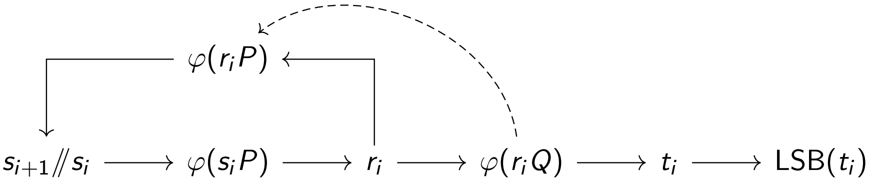
- Goal: Determine s_i for some i.
- $\mathrm{LSB}(t_i)$ only cuts off 16 bits of information.
- Given $\mathrm{LSB}(t_i)$, there are 2^{16} possibilities for t_i.

The attack



- Goal: Determine s_i for some i.
- $\text{LSB}(t_i)$ only cuts off 16 bits of information.
- Given $\text{LSB}(t_i)$, there are 2^{16} possibilities for t_i.
- Find which ones lie on E.

The attack



- Goal: Determine s_i for some i.
- $\mathrm{LSB}(t_i)$ only cuts off 16 bits of information.
- Given $\mathrm{LSB}(t_i)$, there are 2^{16} possibilities for t_i.
- Find which ones lie on E.
- This allows us to find r_iQ. But we want r_iP.

The attack

- Suppose the attacker knows e such that $eQ = P$.

The attack

- Suppose the attacker knows e such that $eQ = P$.
- Say A has x-coordinate t_i. Then $A = r_i Q$.

The attack

- Suppose the attacker knows e such that $eQ = P$.
- Say A has x-coordinate t_i. Then $A = r_i Q$.
- Then $\varphi(eA) = \varphi(er_i Q) = \varphi(r_i eQ) = \varphi(r_i P) = s_{i+1}$

The attack

- Suppose the attacker knows e such that $eQ = P$.
- Say A has x-coordinate t_i. Then $A = r_i Q$.
- Then $\varphi(eA) = \varphi(er_i Q) = \varphi(r_i eQ) = \varphi(r_i P) = s_{i+1}$
- But we still have to solve the DLP to find e, so we're still safe?

Suspicious events

- When NIST published this standard, P and Q were predetermined.
- It was not published how Q was found.
- If an attacker knows $dP = Q$, he can easily compute e such that $eQ = P$.
- It was later revealed that the NSA chose P and Q, and the Snowden leaks suggest that they deliberately inserted a backdoor into this standard.

Questions?