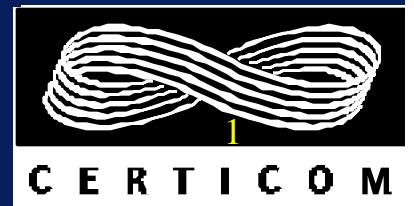


---

# ECDSA: An Enhanced DSA

Don B. Johnson

Alfred J. Menezes



# ECDSA = EC + DSA

---

ECDSA uses an elliptic curve group in the DSA scheme.

# Outline

---

- **Elliptic Curves**
- **DSA definition**
- **DSA/ECDSA mappings**
- **ECDSA definition**
- **Similarities**
- **Advantages**
- **Disadvantages**
- **Summary**

# Background on Elliptic Curves

---

- Elliptic curves have been intensively studied in algebraic geometry and number theory for over one hundred years
- There is enormous literature on elliptic curves
- Elliptic curves played an instrumental role in the recent proof of Fermat's Last Theorem
- Elliptic curves can be used for factoring and proving primality



# ECC Research Activity

---

The implementation of elliptic curve cryptosystems has been studied by many researchers around the world. Numerous research articles have been published, including some by the following:

- T. Beth and F. Schaefer
- J. Chao, K. Tanada and S. Tsujii
- R. Crandall
- B. Kaliski
- N. Koblitz
- K. Koyama and Y. Tsuruoka
- R. Lercier and F. Morain
- W. Meier and O. Staffelbach
- A. Menezes and S. Vanstone
- A. Miyaji
- T. Okamoto, A. Fujioka and E. Fujisaki
- V. Miller



# Abstract Group Definition

- Generalization of arithmetic
- Set of elements in group  $(a, b, c, \dots)$
- Operation  $(*)$  between pairs of elements such that:
  - ④ Closure: For all  $a, b$ ;  $a*b = c$  which is in group
  - ④ Identity: There exists  $i$  such that  $i*a = a*i = a$  for all  $a$
  - ④ Inverses: For all  $a$  there exists  $b$  such that  $a*b = b*a = i$
  - ④ Associativity: For all  $a, b, c$ ;  $(a*b)*c = a*(b*c)$
- Also, if for all  $a, b$ ;  $a*b = b*a$  group is called abelian.

# Elements: Elliptic Curve Points

## Elliptic curves over $\mathbb{Z}_p$

- Let  $p > 3$  be an odd prime, and let  $a$  and  $b$  be integers such that  $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$
- An *elliptic curve* of  $\mathbb{Z}_p$  is defined by an equation

$$E: y^2 = x^3 + ax + b$$

and is the set of all points  $(x,y)$  which satisfy the equation, together with a special element  $\emptyset$ , called the *point at infinity*.

That is

$$E(\mathbb{Z}_p) = \{(x,y) : y^2 = x^3 + ax + b\} \cup \{\emptyset\}$$



# The Group $Z_{23}^*$

- $Z_{23}^*$  consists of the elements of  $Z_{23}$  excluding 0
- $Z_{23}^*$  can be generated by one element

$5^0 = 1$	$5^8 = 16$	$5^{16} = 3$
$5^1 = 5$	$5^9 = 11$	$5^{17} = 15$
$5^2 = 2$	$5^{10} = 9$	$5^{18} = 6$
$5^3 = 10$	$5^{11} = 22$	$5^{19} = 7$
$5^4 = 4$	$5^{12} = 18$	$5^{20} = 12$
$5^5 = 20$	$5^{13} = 21$	$5^{21} = 14$
$5^6 = 8$	$5^{14} = 13$	$5^{22} = 1$
$5^7 = 17$	$5^{15} = 19$	

- The element **5** is called a generator (usually denoted by  $\alpha$ )
- The “group operation” is multiplication



# EC Points Over $\mathbb{Z}_{23}$

- Take  $p = 23$
- Take  $a = 1$  and  $b = 1$ . Note that  
 $27a^3 + 16b^2 = 4 \cdot 1^3 + 16 \cdot 1^2 = 20 \neq 0$  in  $\mathbb{Z}_{23}$
- Consider the elliptic curve  $E$  defined by the equation  
 $y^2 = x^3 + x + 1$
- $E(\mathbb{Z}_{23}) = \{(x,y) : y^2 = x^3 + x + 1\} \cup \{\emptyset\}$

# Solutions to $y^2 = x^3 + x + 1$ Over $\mathbb{Z}_{23}$

(0, 1)	(6, 4)	(12, 19)
(0, 22)	(6, 19)	(13, 7)
(1, 7)	(7, 11)	(13, 16)
(1, 16)	(7, 12)	(17, 3)
(3, 10)	(9, 7)	(17, 20)
(3, 13)	(9, 16)	(18, 3)
(4, 0)	(11, 3)	(18, 20)
(5, 4)	(11, 20)	(19, 5)
(5, 19)	(12, 4)	(19, 18)

$\emptyset$

There are 28 points on this elliptic curve.

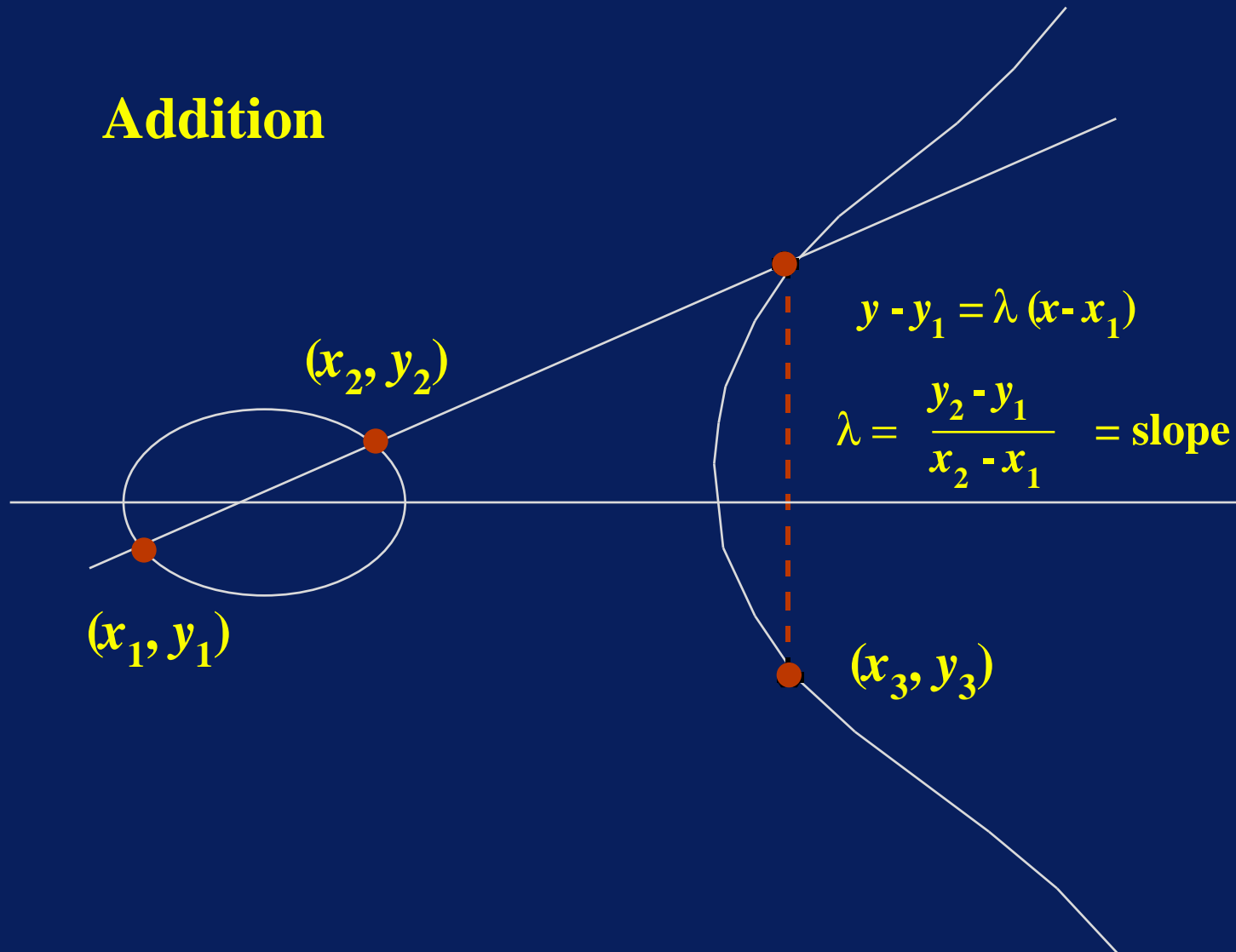


# Group Operation: Addition of Points

- EC addition involves a few arithmetic operations in the underlying field
- Under this operation, the set of elliptic curve points forms a group
- Note that for historical reasons, the operation of an elliptic curve is denoted by “addition”, in contrast to the operation in the group  $\mathbb{Z}_p^*$ , which is denoted by “multiplication”
- Geometric visualization is most intuitive

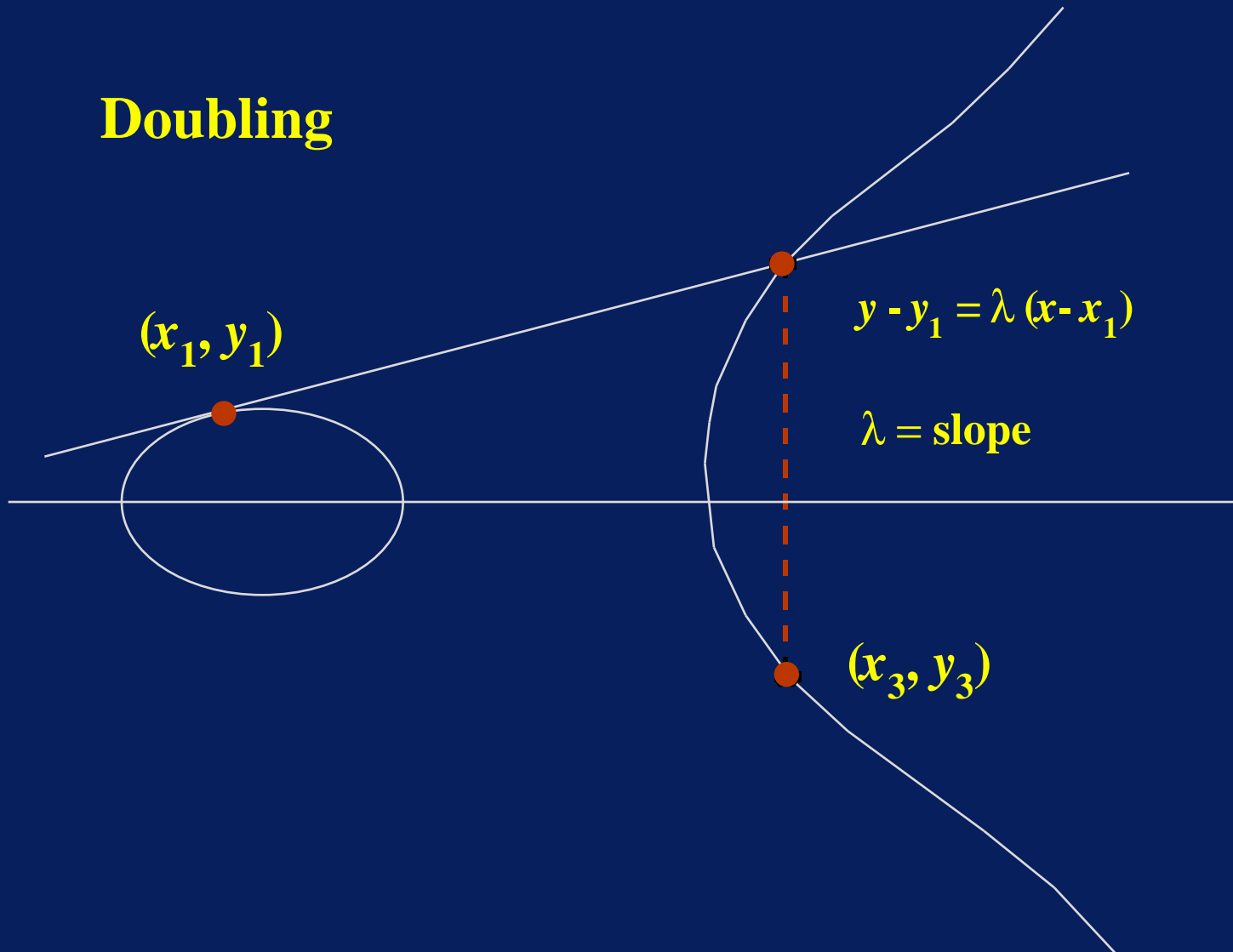
# Elliptic Curve $y^2 = x^3 + ax + b$

## Addition



# Elliptic Curve $y^2 = x^3 + ax + b$

## Doubling



# EC Addition Laws

- $\emptyset + \emptyset = \emptyset$
- $(x, y) + \emptyset = (x, y)$  for all  $(x, y) \in E(\mathbb{Z}_p)$
- $(x, y) + (x, -y) = \emptyset$  for all  $(x, y) \in E(\mathbb{Z}_p)$   
(that is,  $-(x, y) = (x, -y)$ )
- For  $(x_1, y_1) \neq (x_2, y_2)$ ,  $y_1 \neq -y_2$   
 $(x_1, y_1) + (x_2, y_2) = (x_3, y_3)$  where  
$$x_3 = \lambda^2 - x_1 - x_2$$
$$y_3 = \lambda(x_1 - x_3) - y_1$$
$$\lambda = \frac{y_2 - y_1}{x_2 - x_1}$$

# EC Addition Laws

---

- For  $(x_1, y_1) \in E(\mathbb{Z}_p)$ ,  
 $2(x_1, y_1) = (x_3, y_3)$  where  
$$x_3 = \lambda^2 - 2x_1$$
$$y_3 = \lambda(x_1 - x_3) - y_1$$

$$\lambda = \frac{3x_1^2 + a}{2y_1}$$

# Example of Addition in $E(\mathbb{Z}_{23})$

Let  $P_1 = (3, 10), P_2 = (9, 7),$

$$P_1 + P_2 = (x_3, y_3).$$

$$\lambda = \frac{7-10}{9-3} = \frac{-3}{6} = \frac{-1}{2} = 11 \in E(\mathbb{Z}_{23})$$

$$x_3 = 11^2 - 3 - 9 = 6 - 3 - 9 = 17$$

$$\begin{aligned} y_3 &= 11(3 - (-6)) - 10 = 11(9) - 10 \\ &= 89 = 20. \end{aligned}$$

Therefore  $P_1 + P_2 = (17, 20)$

# Example of Doubling in $E(\mathbb{Z}_{23})$

Let  $P_1 = (3, 10)$

$2P_1 = (x_3, y_3),$

$$\lambda = \frac{3(3^2) + 1}{20} = \frac{5}{20} = \frac{1}{4} = 6$$

$$x_3 = 6^2 - 6 = 30 = 7,$$

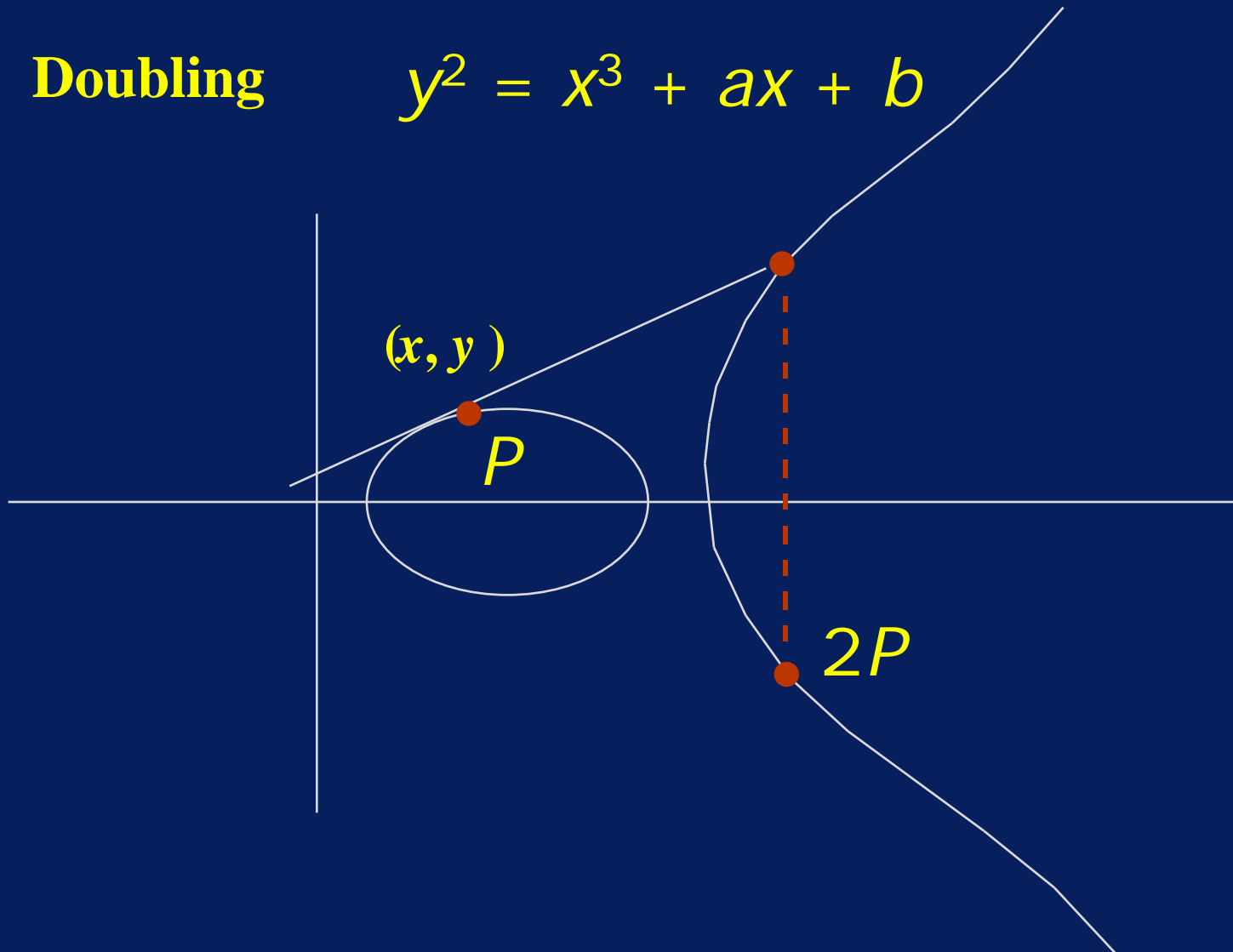
$$y_3 = 6(3 - 7) - 10 = -24 - 10 = 12$$

Therefore  $2P_1 = (7, 12)$

# ECDLP - A Hard Problem

**Doubling**

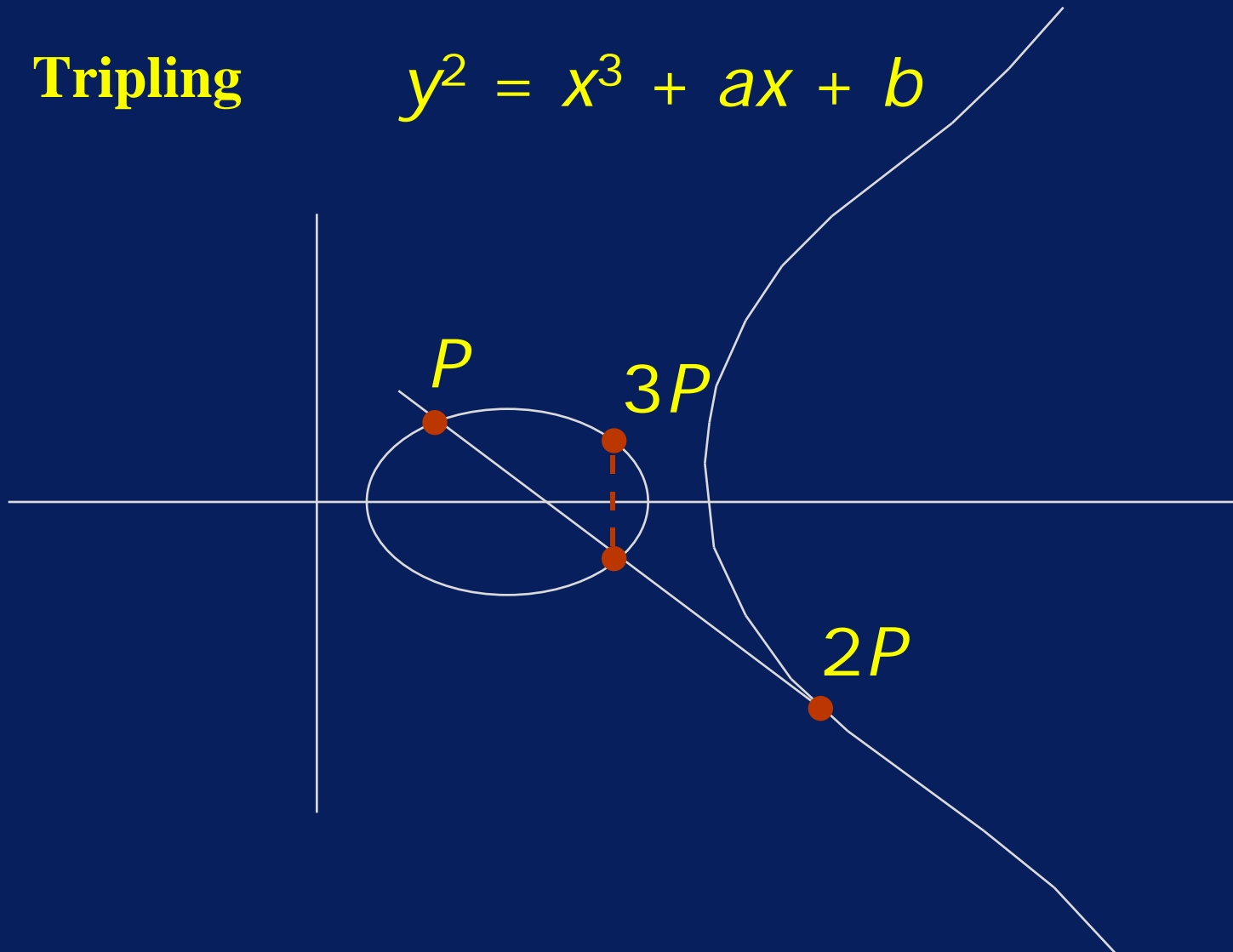
$$y^2 = x^3 + ax + b$$



# ECDLP - A Hard Problem

**Tripling**

$$y^2 = x^3 + ax + b$$



# ECDLP - A Hard Problem

---

Now keep adding the point “ $P$ ” to itself about:

10,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000 times

or  $10^{55}$  times (this is the implementation challenge)

You will end up on some point on the curve.

The hard problem is: given the starting point and the ending point, how many times did I add the starting point to itself?

# Difficulty of ECDLP

---

The difficulty of the elliptic curve discrete logarithm problem (ECDLP) has been studied for over 11 years by various researchers including the following:

- Adleman, DeMarrais and Huang
- G. Frey and H. Rück
- N. Koblitz
- A. Menezes, T. Okamoto and S. Vanstone
- V. Miller

No significant weaknesses have been found.

# Elliptic Curves over $GF(2^m)$

---

- The discussion so far has been about elliptic curves over  $\mathbb{Z}_p$
- The theory extends in a natural way to elliptic curves over the finite field  $GF(2^m)$
- These fields are advantageous over  $\mathbb{Z}_p$  because the arithmetic in the underlying finite field  $GF(2^m)$  can be implemented more efficiently than the arithmetic in the finite field  $\mathbb{Z}_p$

---

# DSA Definition

# DSA Key Pair Generation

- Let  $p$  be a large prime (512 to 1024 bits) such that  $p-1$  has a large (160 bit) prime divisor  $q$
- Let  $g$  be a generator of the unique subgroup of  $GF(p)$  of order  $q$
- Alice picks a random integer  $x$ ,  $0 < x < q$ , and computes  $y = g^x \bmod p$
- Alice's public key is  $y$  and her private key is  $x$

# DSA Signature Generation

To sign a message  $m$ , Alice does the following:

- Picks a random integer  $k$ ,  $0 < k < q$
- Computes  $r = (g^k \bmod p) \bmod q$
- Solves the congruence

$$s = k^{-1} \{H(m) + xr\} \bmod q$$

(where H is SHA-1 hash function)

The signature for the message  $m$  is the pair of integers  $(r, s)$ .

# DSA Signature Verification

To verify Alice's signature  $(r, s)$  for a message  $m$ , Bob does the following:

- Computes  $w = s^{-1} \bmod q$
- Computes  $u1 = H(m)w \bmod q$  and  $u2 = rw \bmod q$
- Computes  $v = ((g^{u1} y^{u2}) \bmod p) \bmod q$
- Verifies that  $v = r$

# DSA/ECDSA Group Mappings

- Elements: integers 1 to  $p-1$
  - Operation: multiplication of integers
  - Multiplication:  $gh$
  - Inverse:  $1/g$
  - Division:  $g/h$
  - Exponentiation:  $g^a$
  - DLP: Given  $g$  and  $h = g^a \bmod p$ , find  $a$
- Elements: EC points
  - Operation: addition of points
  - Addition:  $P + Q$
  - Inverse:  $-P$
  - Subtraction:  $P - Q$
  - Multiple of a point:  $aP$
  - ECDLP: Given  $P$  on EC and  $Q = aP$ , find  $a$

# DSA/ECDSA Parameter Mappings

- $g$  - generator of subgroup
- $q$  - order of  $g$
- $x$  - private key
- $y$  - public key
- $p$  &  $q$  primes
- $g$  is element of order  $q$
- The group elements are the powers of  $g$
- $P$  - generator of EC subgroup
- $n$  - order of  $P$
- $d$  - private key
- $Q$  - public key
- $E$  is EC over  $GF(q)$
- $P$  is point of order  $n$
- The group elements are the multiples of  $P$

# ECDSA Key Pair Generation

---

- Let  $E$  be an elliptic curve over  $\mathbb{Z}_p$  or  $GF(2^m)$
- Let  $P$  be a point of order  $n$ , a large prime
- Alice picks a statistically unique and unpredictable integer  $d$ ,  $1 < d < n-1$ , and computes the point  $Q = dP$
- Alice's public key is  $Q$ , her *private key* is  $d$

# ECDSA Signature Generation

To sign a message  $m$ , Alice does the following:

- Picks a statistically unique and unpredictable integer  $k$  such that  $1 < k < n-1$
- Computes  $R = kP$ . Let  $r$  be the x-coordinate of  $R$
- Solves the congruence

$$s = k^{-1} \{ H(m) + dr \} \bmod n$$

(where H is SHA-1)

The signature for the message  $m$  is the pair of integers  $(r, s)$



# ECDSA Signature Verification

To verify Alice's signature  $(r, s)$  for a message  $m$ , Bob does the following:

- Verify  $0 < r < n$  and  $0 < s < n$ , or reject signature.
- Computes  $w = s^{-1} \bmod n$
- Computes  $u_1 = H(m)w \bmod n$  and  $u_2 = rw \bmod n$
- Computes the point  $V = u_1P + u_2Q$ .  
Let  $v$  be the  $x$ -coordinate of  $V$
- Verifies that  $v = r$

# Similarities

---

- Both are based on ElGamal signatures, both use the same signing equation
- Both use SHA-1 as the hash function
- In both, the values that are difficult to generate are system parms (DSA  $p, q, g$  and ECDSA  $q, a, b, P$  and  $n$ ) which are public and easily verifiable
- Both use a normative seeded function to help ensure system parameters were not generated to meet some hidden criteria

---

# Advantages

# Honey, I Shrunk The Keys

Strength/key size comparison:

<b>RSA/DSA Key Size</b>	<b>Comparable EC Key Size</b>	<b>Time to Break (MIPS Years)</b>	<b>Key Size Ratio</b>
512	106	$10^4$	4.6
768	132	$10^8$	5.6
1024	160	$10^{12}$	6.4
2048	211	$10^{20}$	9.4
5120	320	$10^{36}$	16.0

Results are based on the factoring estimates from A. Odlyzko's paper, "The Future of Integer Factorization", CryptoBytes Vol. 1(2), Summer 1995.

# Security of EC versus DSA/RSA

- EC gives the most security per bit of any known public-key scheme.
- The ECDLP problem appears to be much more difficult than the integer factorization problem or the discrete logarithm problem in  $\mathbb{Z}_p^*$
- The strength of EC grows much faster with key size increases than does the strength of RSA or DSA

# Remarks on Security

- Estimated time to factor 1024-bit numbers:

$3 \times 10^{11}$  MIPS years

(Estimates\* by A. Odlyzko based on Lenstra's Number Field Sieve (NFS) Implementation)

- Assume a 1 MIPS machine performs 40,000 elliptic curve operations. Estimated time to compute logs in a suitably chosen elliptic curve over  $F_{2^{160}}$  is  $10^{12}$  MIPS years

\* Odlyzko, Andrew, "The Future of Integer Factorization", CryptoBytes, Summer 1995, p. 5.

# Honey, I Shrunk The Certificates

Short keys result in short signatures and certificates

	<b>RSA</b>	<b>DSA</b>	<b>EC</b>
Public Key Size	1024 bits	1024 bits	160 bits
Signature	1024 bits	320 bits	320 bits
X.509 Certificate	( <b>2048</b> + $K$ ) bits	( <b>1344</b> + $K$ ) bits	( <b>480</b> + $K$ ) bits

where  $K$  is the number of common bits in each certificate type.

# Honey, I Shrunk The Timings

- DSA
  - Operation:  $g^k \bmod p$
  - $p$  is 1024 bits
  - $k$  is 160 bits
  - Expect 160 squarings and 80 multiplications
  - 240 1024-bit field multiplications
- ECDSA
  - Operation:  $kP$
  - $E(\mathbb{Z}_p)$  where  $p$  is 160 bits
  - $k$  is 160 bits
  - Expect 160 EC doublings and 80 EC additions
  - each at 5 field multiplications
  - 1200 160-bit field multiplications
  - Multiplication is 41 times faster
  - Net: 8 times faster



# 99 44/100 % Pure?

---

- In DSA signature generation, there is an optional bounds check that if not done could result in a signature failing to verify. In ECDSA, the analogous bounds check is mandatory.
- In DSA, the prime test is probabilistic. In ECDSA, due to EC technology, there is an optional deterministic prime test, if one wants to spend the MIPS.
- In DSA, the private key  $x$  and per-signature value  $k$  are “random”. In ECDSA they are defined to be “statistically unique and unpredictable”. This allows high security use to filter for duplicate values.



# How Do I Know She Really Loves Me?

---

- In ECDSA an explicit method is given to verify the system parms.
- At Crypto '96, Serge Vaudenay presented an attack on DSA that allowed an adversary to generate one bogus signature if he could create system parms. This was based on the insight that the DSA hash was really  $\text{SHA-1 mod } q$ . In ECDSA, if  $n > 2^{160}$ , then Vaudenay's weakness does not exist.

# 5 EC Pieces

---

## X9.62

- Over 60 pages of supplementary text
- Complete worked out examples for 5 key sizes: 155, 158, 209, 210, and 239 bits.
- 20 elliptic curves for each of 5 key sizes
- Suggested primes

# Requirement Clarification

---

- DSA states that the integrity of signed data is dependent upon the prevention of unauthorized disclosure, modification, substitution, insertion, and deletion of the private key.
- ECDSA adds the prevention of unauthorized use to the above.

# Subtle Advantage/Difference

---

- In DSA, there is a possible public key for which computing log takes no modular exponentiation:  $g$ , this corresponds to a  $k$  value of 1.
- In ECDSA the corresponding special points are  $P$  and  $-P$  which take no EC calculations to compute, for which  $k$  is 1 or  $n-1$ . These 2 points are excluded, the valid range for  $k$  is 2 to  $n-2$  inclusive.

# Disadvantages

---

# ECDSA Summary

---

- Based on a harder problem than DSA
- Shorter keys
- Shorter certificates
- Faster
- Based on DSA: Should facilitate understanding and acceptance
- Cleans up some minor loose ends in DSA
- Net: A better DSA