

RSA® Conference 2019

San Francisco | March 4–8 | Moscone Center

BETTER.

SESSION ID: STR1-T08

A General Introduction to Modern Cryptography

Josh Benaloh

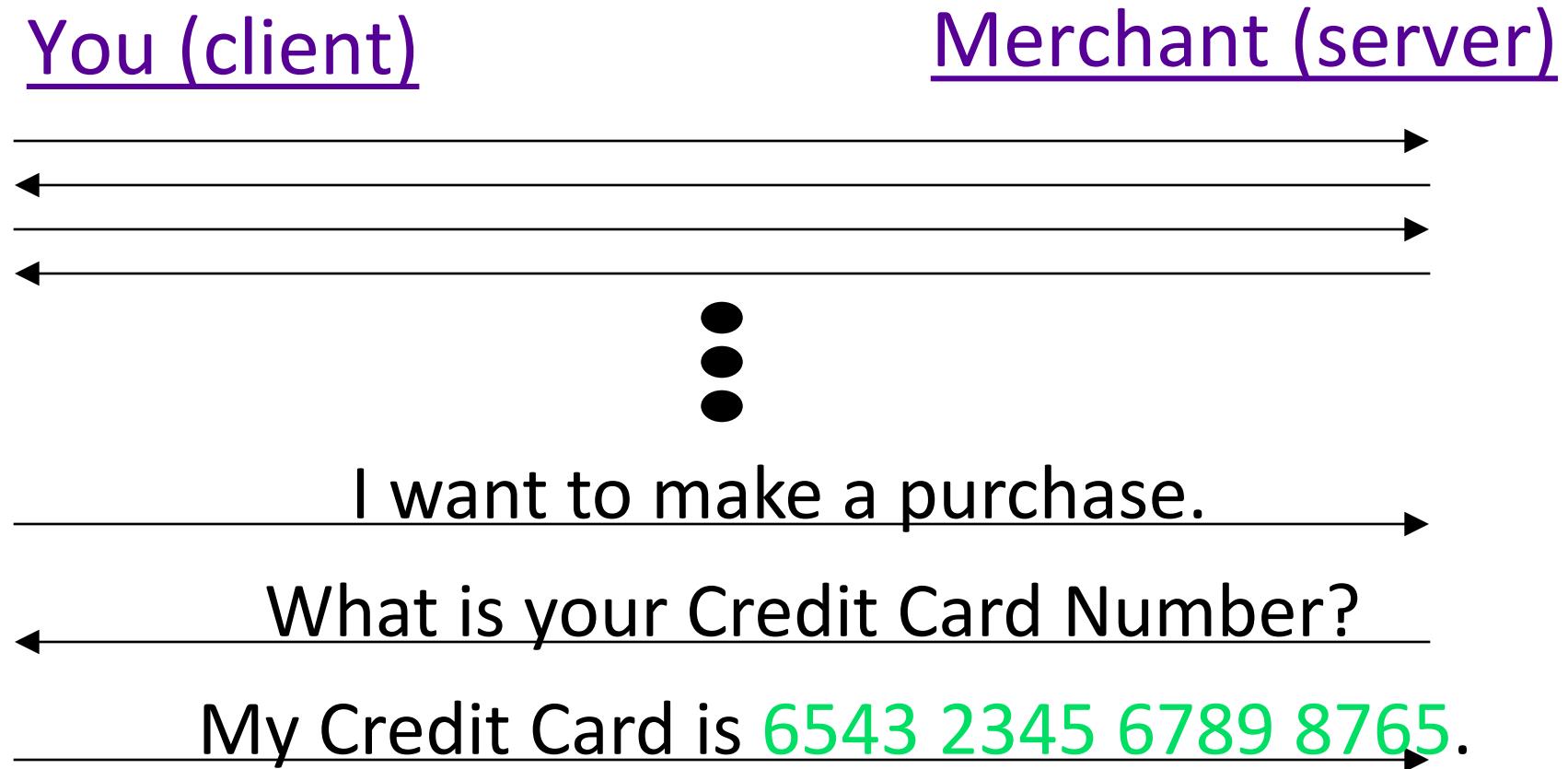
Senior Cryptographer
Microsoft Research

#RSAC

Cryptography Basics on the Internet

- The Internet was *not* designed for security.
- Sending data via the Internet is like sending post cards through the mail ...
...when you don't trust the Post Office.

A Typical Internet Transaction





Can we at least protect the credit card number so that it won't be revealed to anyone except the intended merchant?

Kerckhoffs's Principle (1883)

The security of a cryptosystem should depend only on the key.

You should assume that attackers know everything about your system *except* the key.

Symmetric Encryption

- If the client has a pre-existing relationship with the merchant, the two parties may have a shared secret key K – known only to these two.
- User encrypts private data with key K .
- Merchant decrypts data with key K .

Requirements for a Key

A key should be really, *really* hard to guess ...

- Even if you have a lot of time.
- Even if you have a lot of computational resources.
- Even if you have a lot of samples that use the key.

PINs, Passwords, & Keys

Informally ...

- A *PIN* is a **4-6** digit speed bump.
- A *password* is a short, user-chosen, usually guessable selection from a small dictionary.
- A *key* is an unguessable, randomly chosen string – usually at least **128** bits.

On-Line Defenses

On-line attacks can be mitigated.

- Rates can be controlled.
- Counts can be limited.
- Real-time monitoring can detect unusual access patterns.

Off-Line Attacks

- Encryption keys are subject to *off-line* attacks.
- An attacker can search a space of roughly 2^{64} values.
 - All PINs of fewer than 20 digits
 - All passwords of fewer than 14 lower case letters
 - All alphanum passwords of fewer than 12 characters
 - All printable passwords of fewer than 10 characters

Off-Line Attacks

- Don't even think about using user-chosen passwords as encryption keys.
- Don't even think about using keys derived deterministically from user-chosen passwords.
- Given the ciphertext, an attacker can do a (guided) exhaustive search through the space to find the password.

Modern Symmetric Ciphers

Private-key (symmetric) ciphers are usually divided into two classes.

- Stream ciphers
- Block ciphers

Modern Symmetric Ciphers

Private-key (symmetric) ciphers are usually divided into two classes.

- Stream ciphers
- Block ciphers

Stream Ciphers

RC4, A5/1, SEAL, SPRITZ, etc.

- Use the key as a *seed* to a pseudo-random number-generator (PRNG).
- Take the stream of output bits from the PRNG and XOR it with the plaintext to form the ciphertext.

The XOR Function

\otimes	0	1
0	0	1
1	1	0

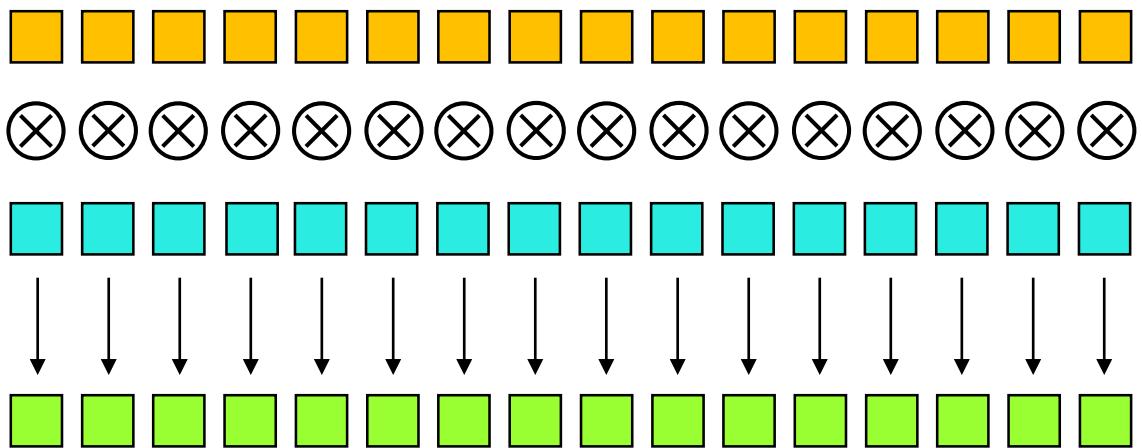
Each of the two input bits indicates whether or not the other bit is flipped.

One-Time Pad

Plaintext:

Key:

Ciphertext:



Stream Ciphers

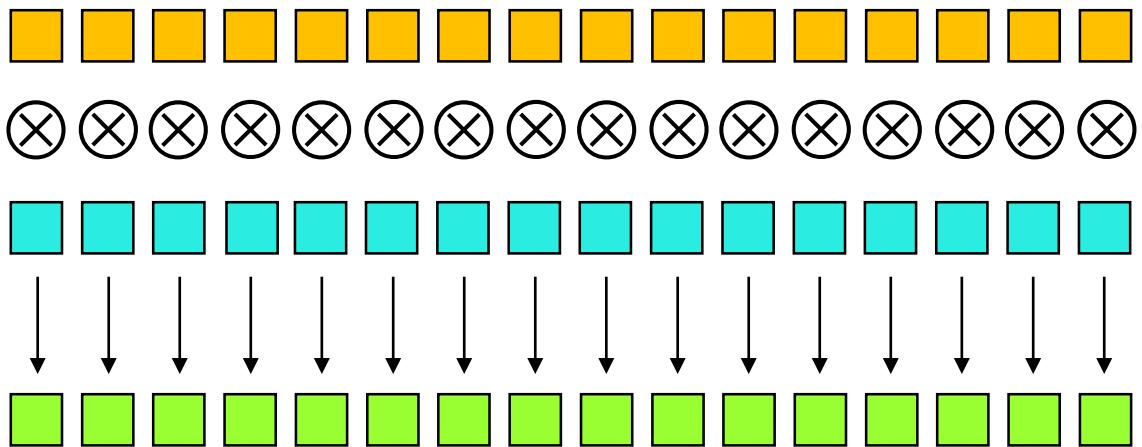
Replace the long random key with a *pseudo-random* number generated by using a short random key as a seed.

One-Time Pad

Plaintext:

Key:

Ciphertext:



Stream Cipher Encryption

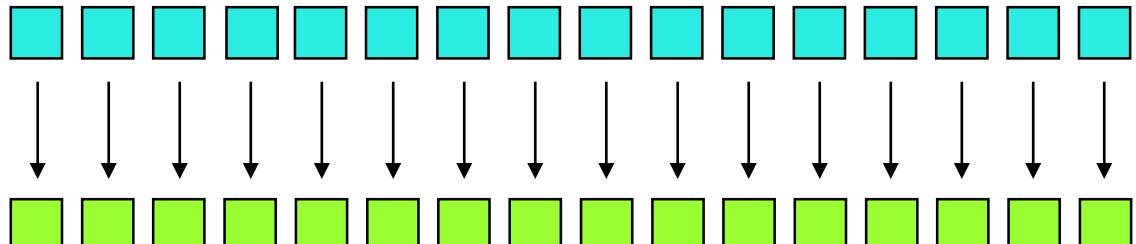
Plaintext:



PRNG(seed):

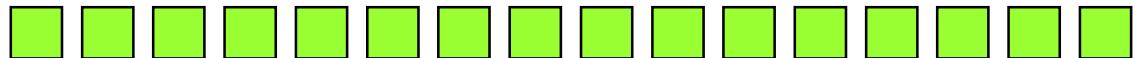


Ciphertext:



Stream Cipher Decryption

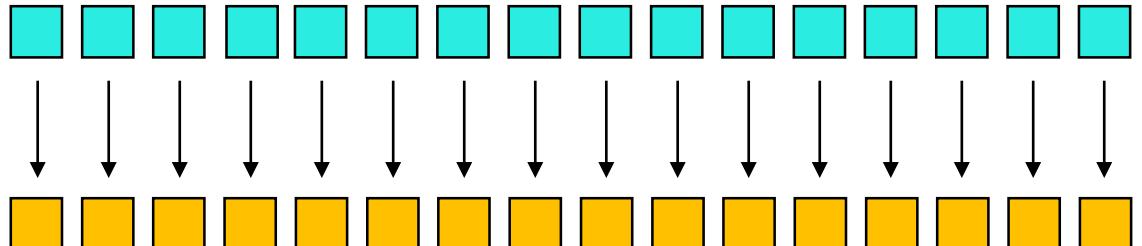
Ciphertext:



PRNG(seed):



Plaintext:



A PRNG: Alleged RC4

Initialization

$S[0..255] = 0, 1, \dots, 255; j = 0$

$K[0..255] = Key, Key, Key, \dots$

for $i = 0$ to 255

$j = (j + S[i] + K[i]) \bmod 256$

swap $S[i]$ with $S[j]$

A PRNG: Alleged RC4

Iteration

$i = (i + 1) \text{ mod } 256$

$j = (j + S[i]) \text{ mod } 256$

swap $S[i]$ with $S[j]$

$t = (S[i] + S[j]) \text{ mod } 256$

Output $S[t]$

Some Good Properties

- Stream ciphers are typically very fast.
- Stream ciphers can be very simple.
- The same function is used for encryption and decryption.

Stream Cipher Insecurity

If two plaintexts are *ever* encrypted with the same key stream

$$\begin{aligned}C_1 &= K \otimes P_1 \\C_2 &= K \otimes P_2\end{aligned}$$

an attacker can easily compute

$$C_1 \otimes C_2 = P_1 \otimes P_2$$

from which P_1 and P_2 can usually be teased apart easily.

Stream Cipher Encryption

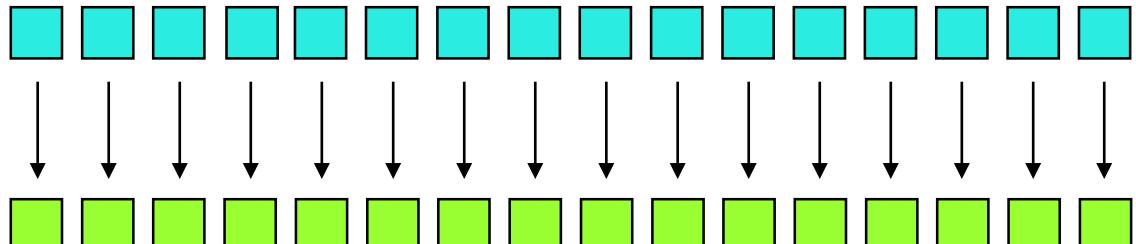
Plaintext:



PRNG(seed):



Ciphertext:



Stream Cipher Integrity

- It is easy for an adversary (even one who can't decrypt the ciphertext) to alter the plaintext in a known way.

Stream Cipher Integrity

- It is easy for an adversary (even one who can't decrypt the ciphertext) to alter the plaintext in a known way.

Bob to Bob's Bank:

Please transfer \$1,000,002.00 to the account
of my good friend Alice.

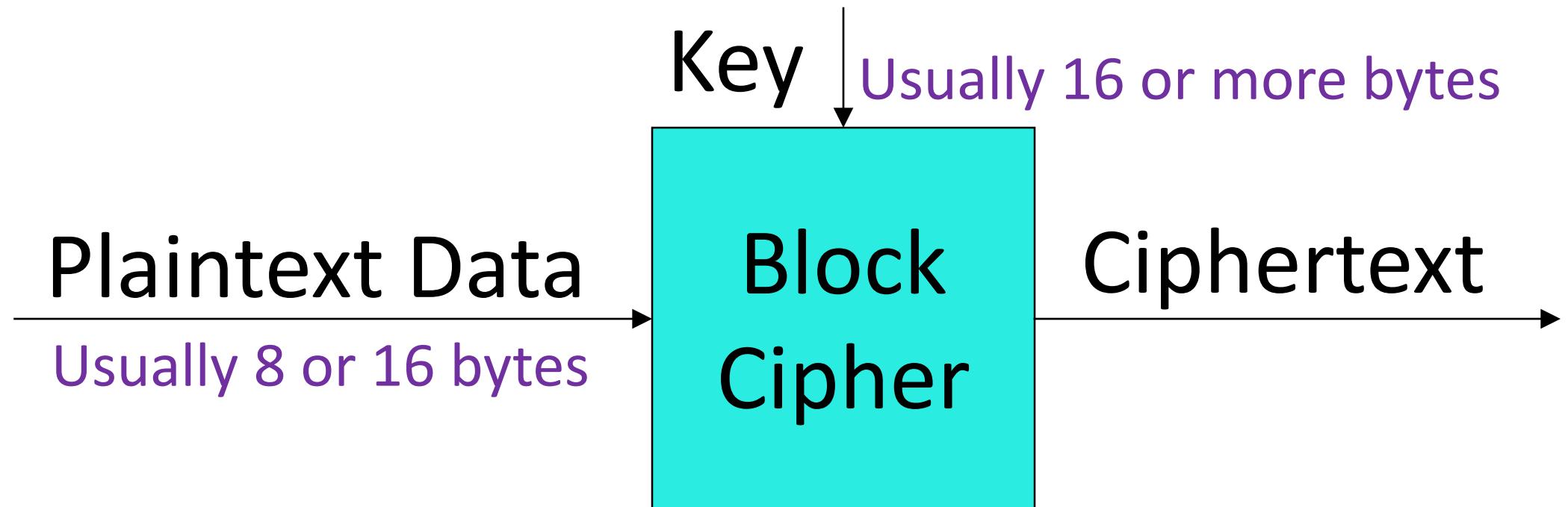
Modern Symmetric Ciphers

Private-key (symmetric) ciphers are usually divided into two classes.

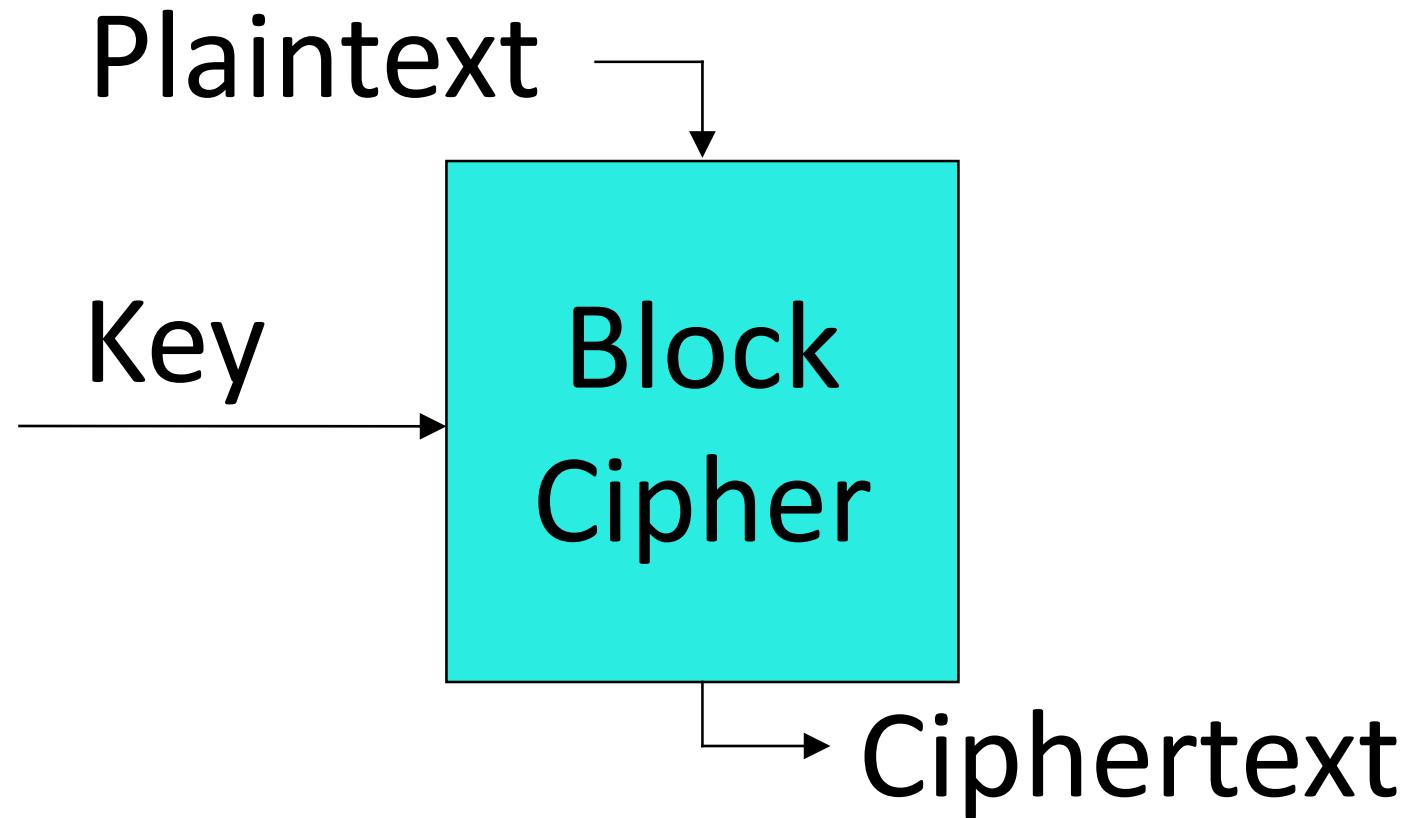
- Stream ciphers
- Block ciphers

Block Ciphers

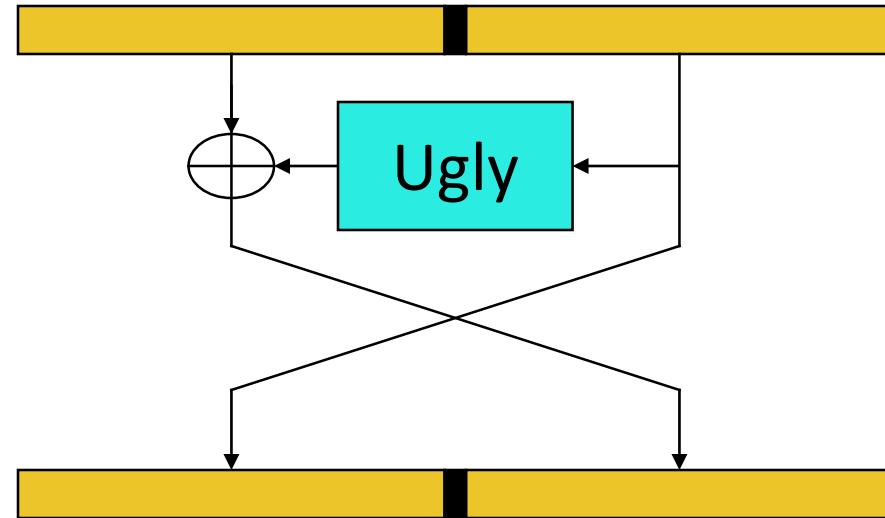
AES, DES, 3DES, Twofish, etc.



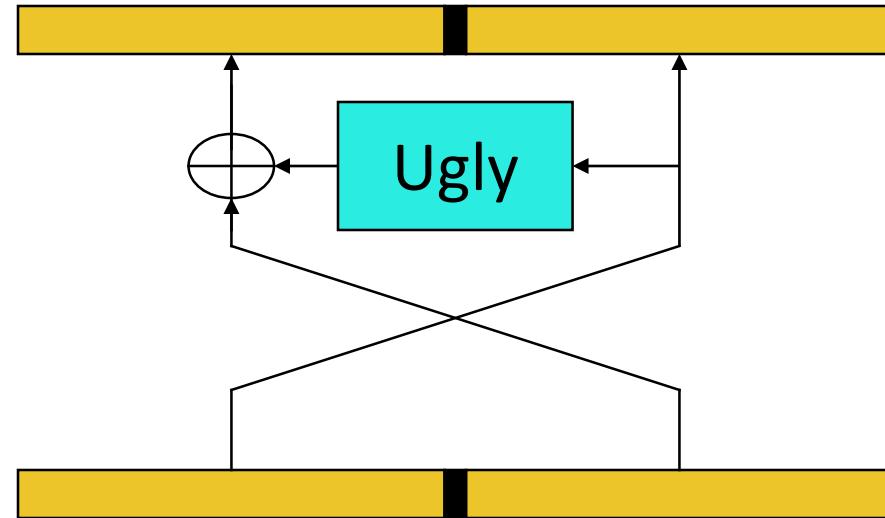
How to Build a Block Cipher



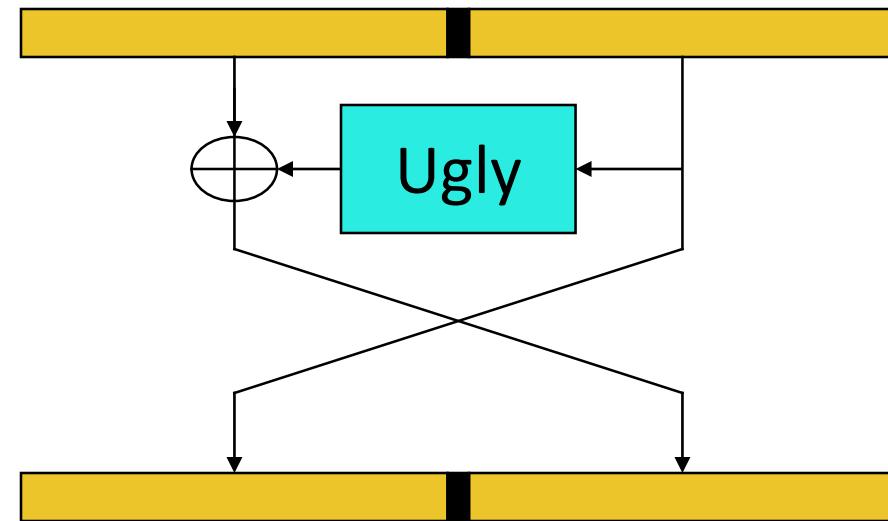
Feistel Ciphers



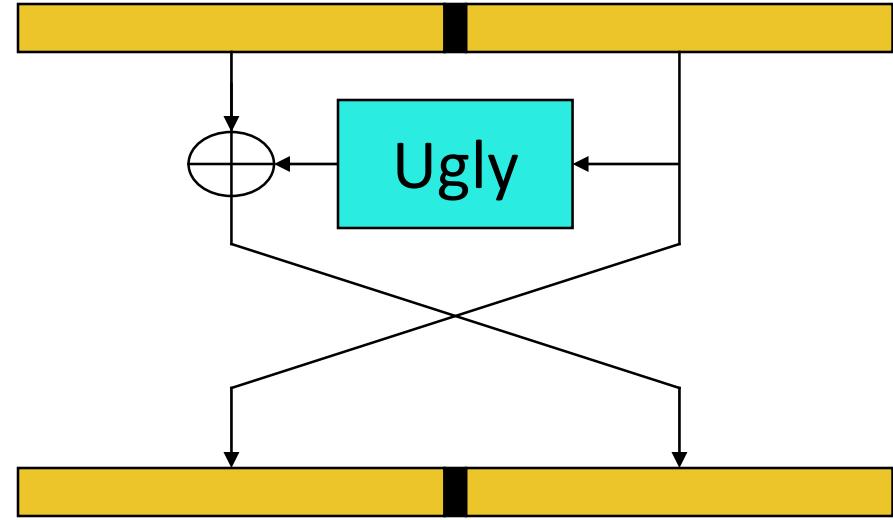
Feistel Ciphers



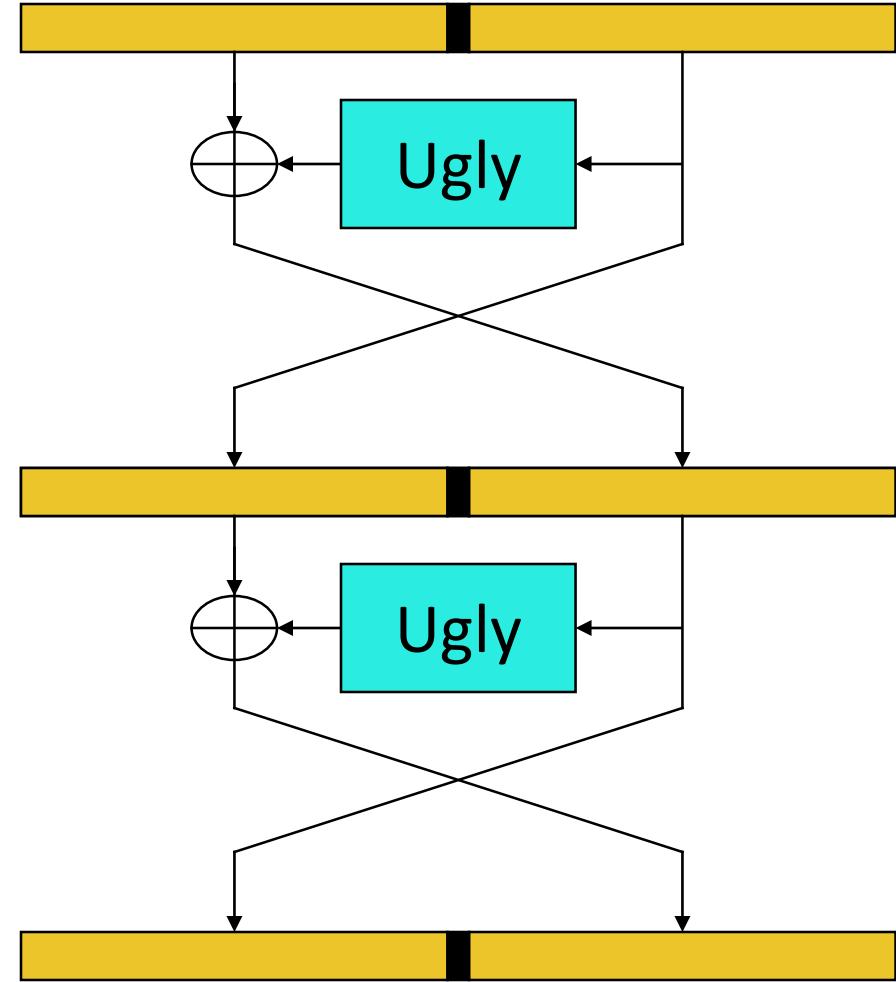
Feistel Ciphers



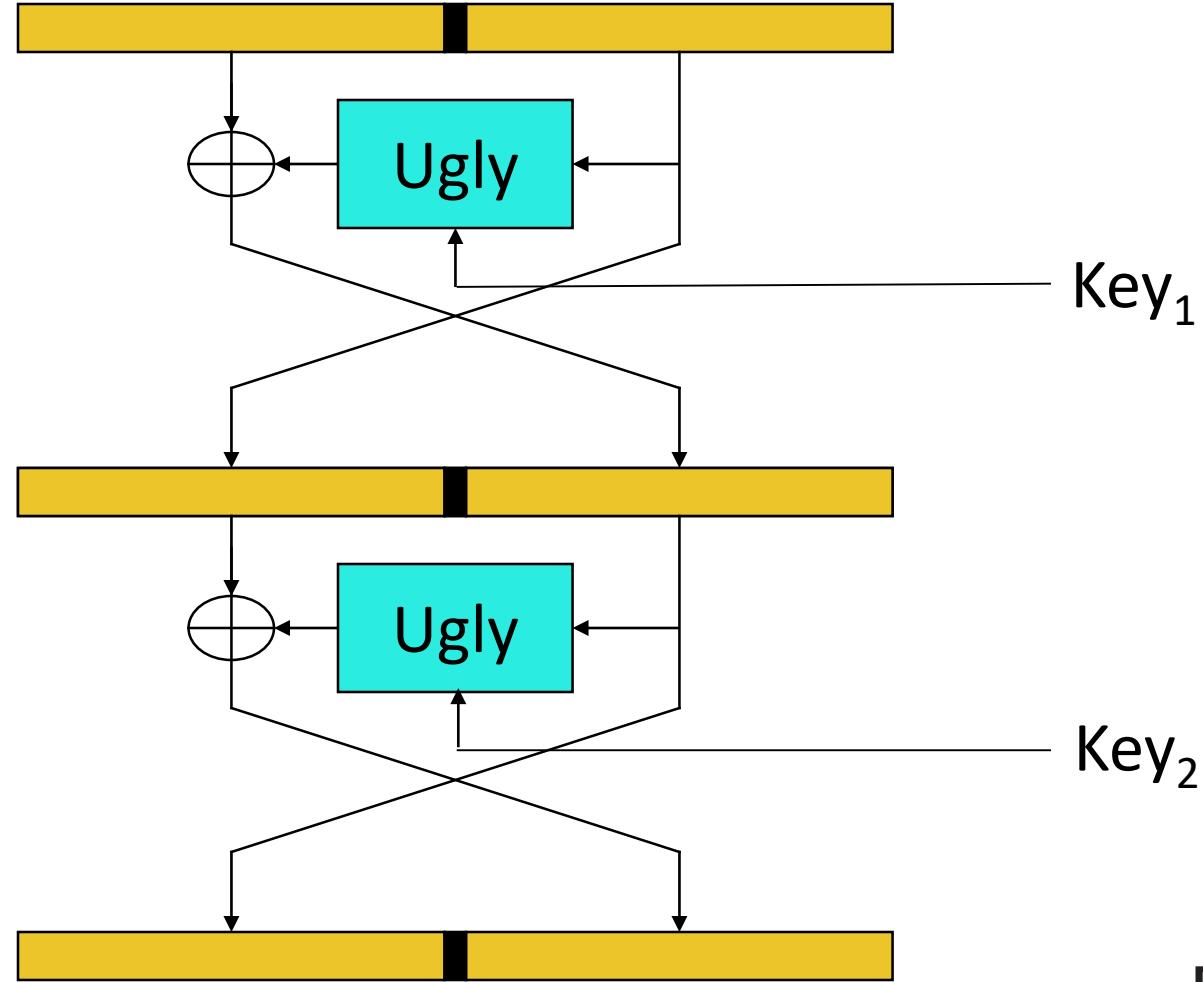
Feistel Ciphers



Feistel Ciphers



Feistel Ciphers

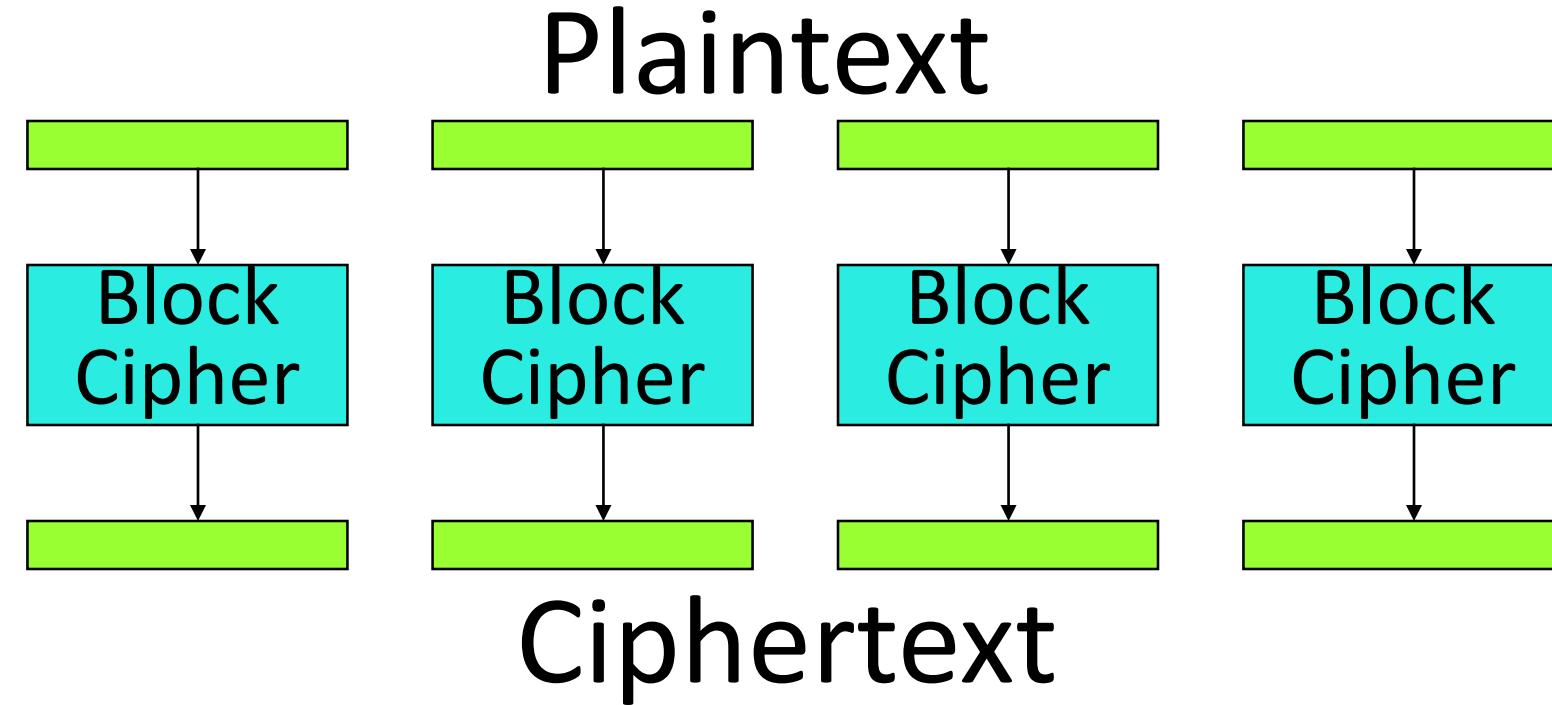


Feistel Ciphers

- Typically, Feistel ciphers are iterated for about 10-16 rounds.
- Different “sub-keys” are used for each round.
- Even a weak round function can yield a strong Feistel cipher if iterated sufficiently.

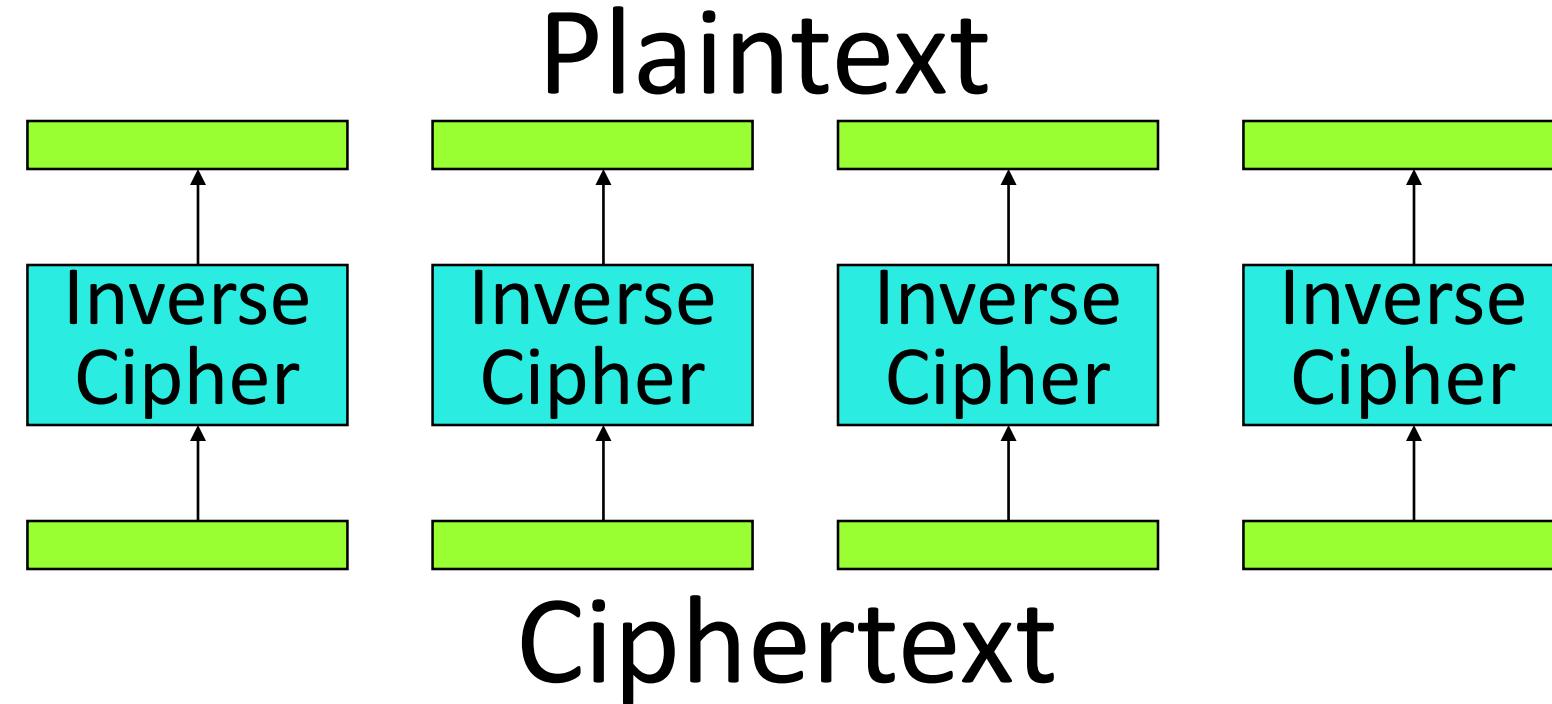
Block Cipher Modes

Electronic Code Book (ECB) Encryption:



Block Cipher Modes

Electronic Code Book (ECB) Decryption:



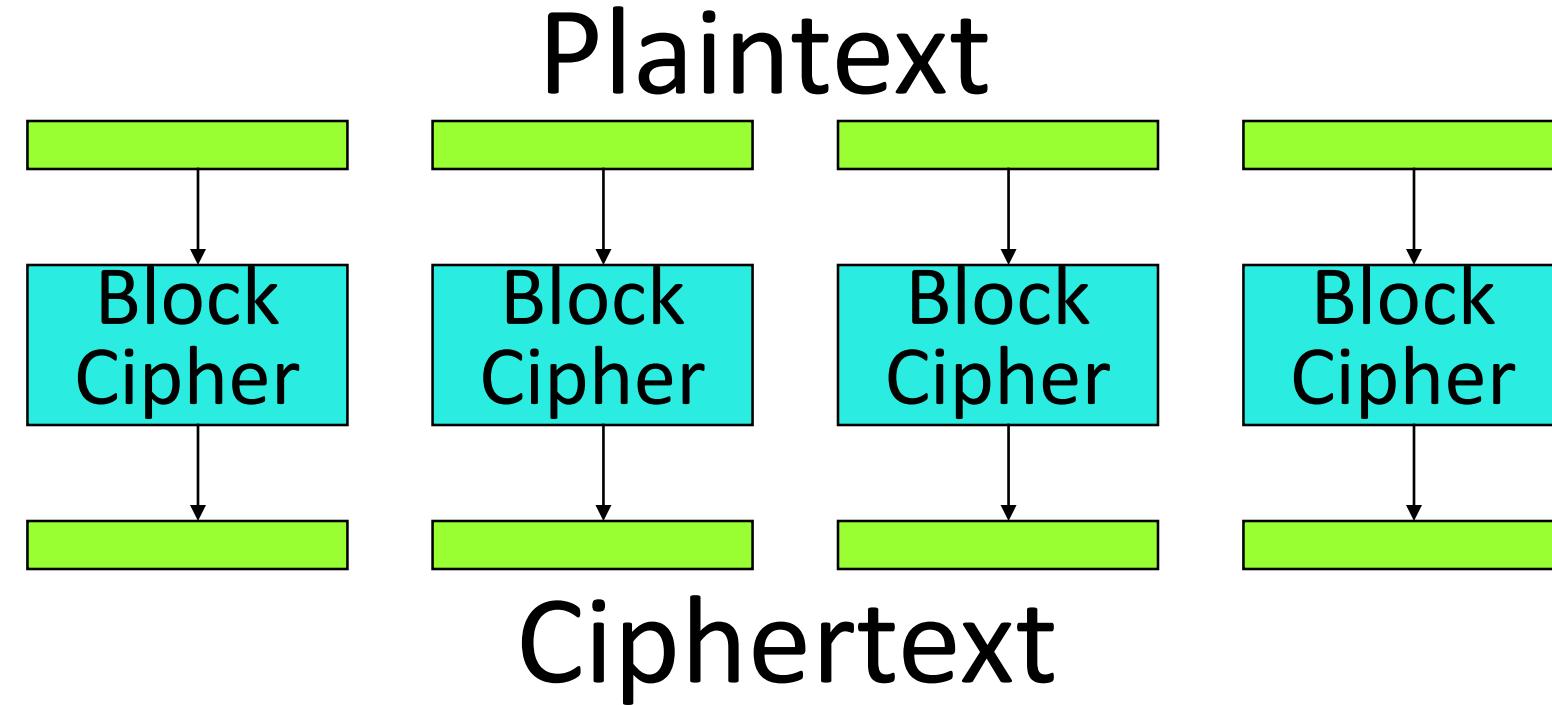
Block Cipher Integrity

With ECB mode, identical blocks will have identical encryptions.

This can enable replay attacks as well as re-orderings of data. Even a passive observer may obtain statistical data.

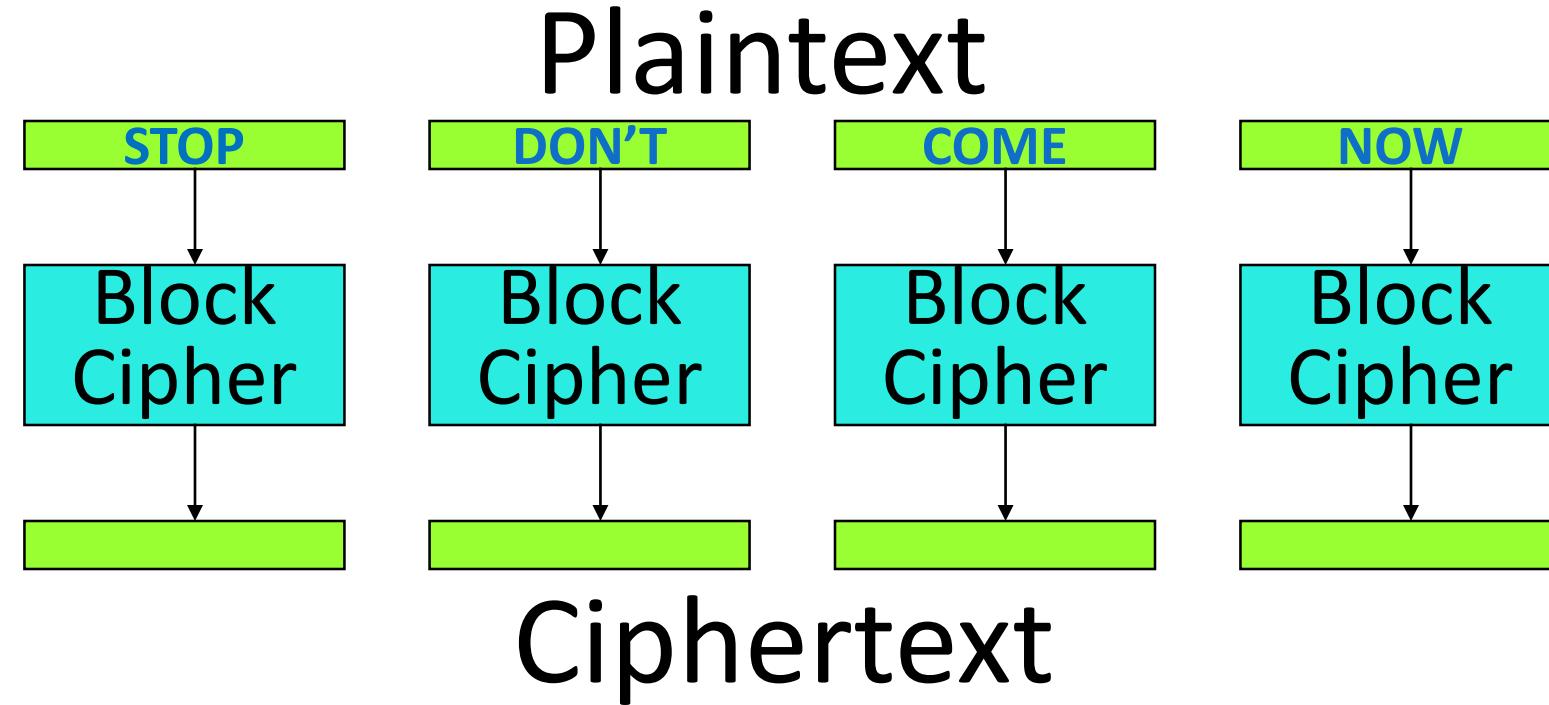
Block Cipher Modes

Electronic Code Book (ECB) Encryption:



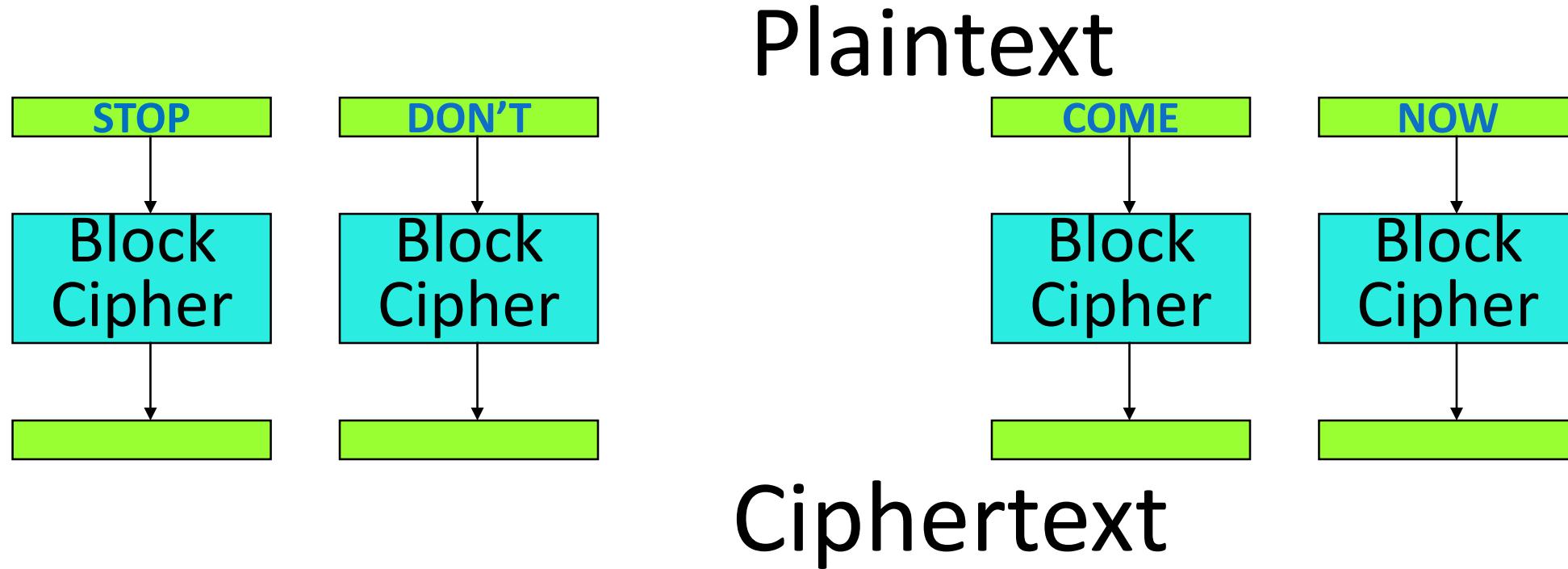
Block Cipher Modes

Electronic Code Book (ECB) Encryption:



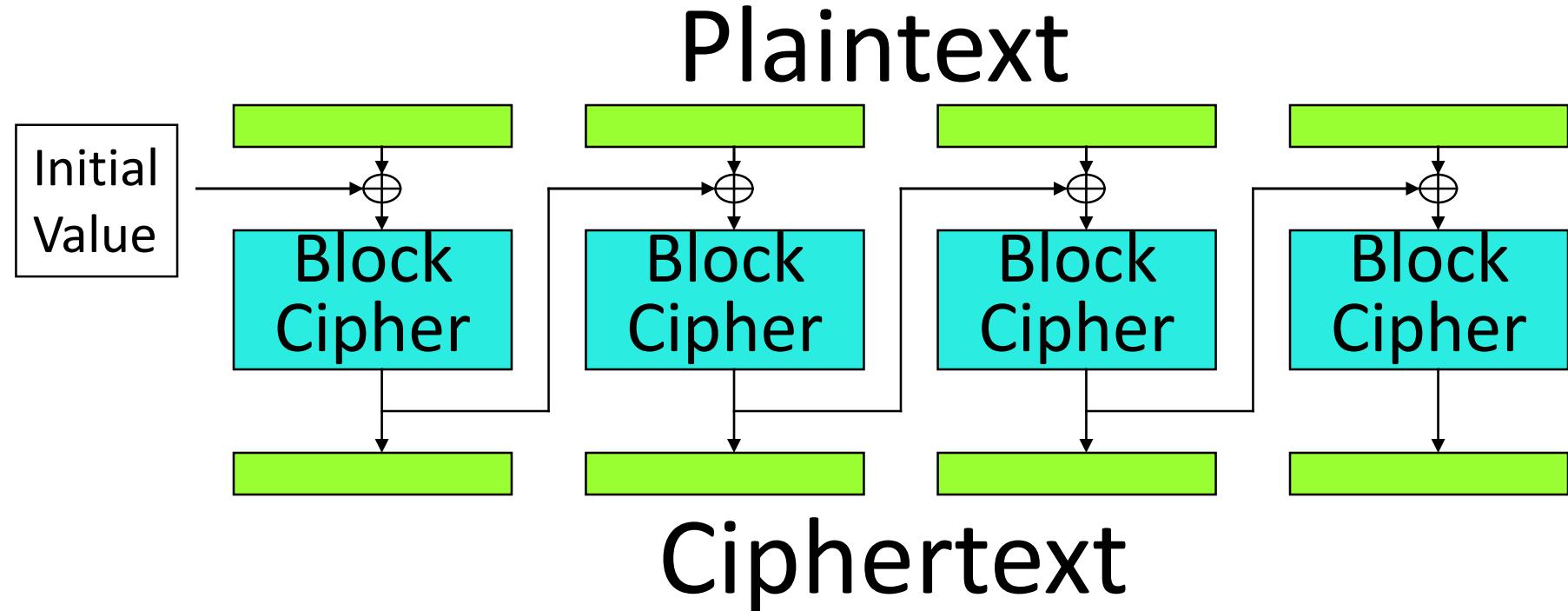
Block Cipher Modes

Electronic Code Book (ECB) Encryption:



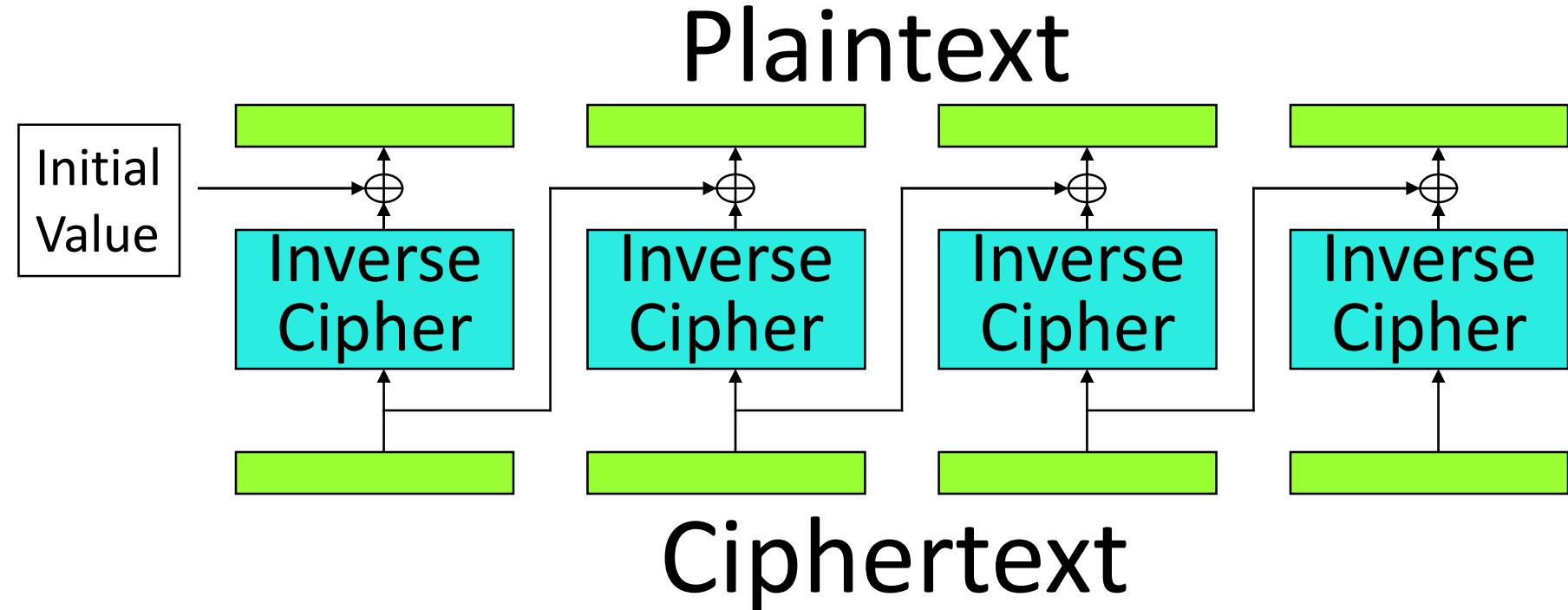
Block Cipher Modes

Cipher Block Chaining (CBC) Encryption:



Block Cipher Modes

Cipher Block Chaining (CBC) Decryption:

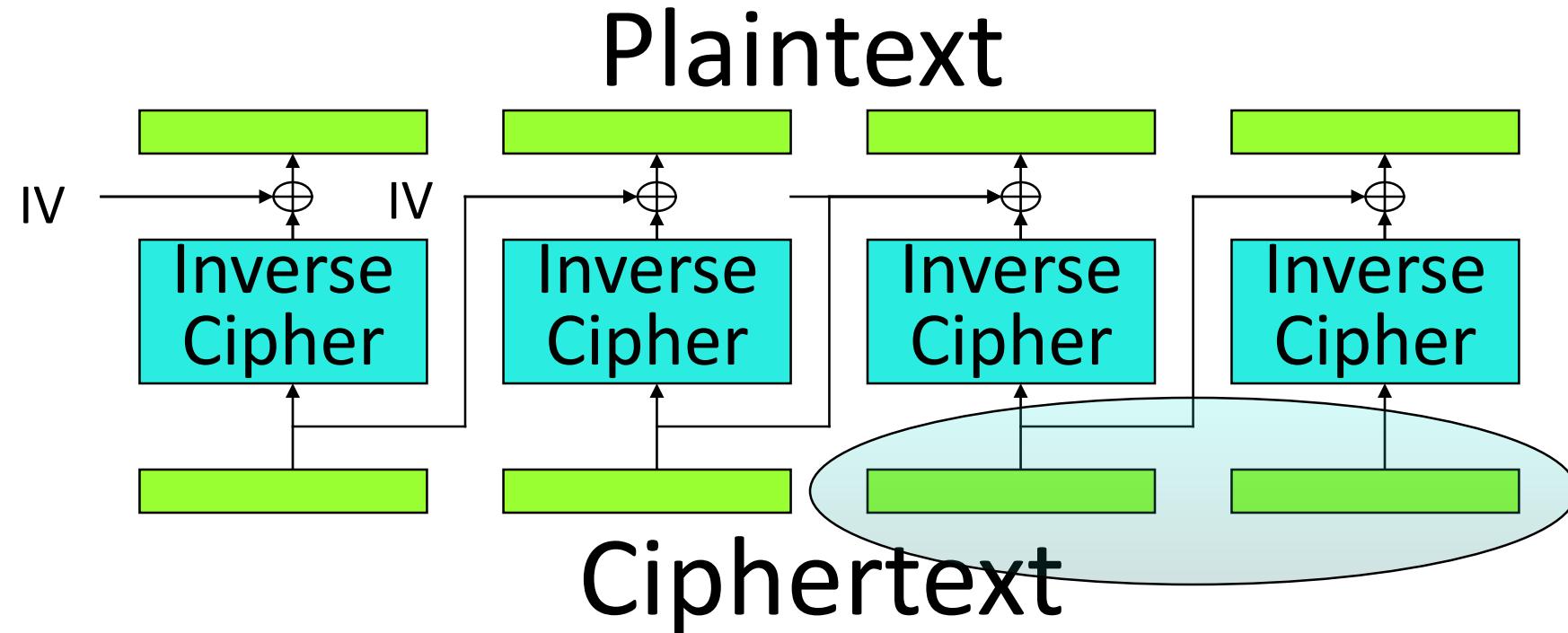


Some CBC-Mode Myths

- I can't use CBC mode because I need random-access decryption.

Block Cipher Modes

Cipher Block Chaining (CBC) Decryption:

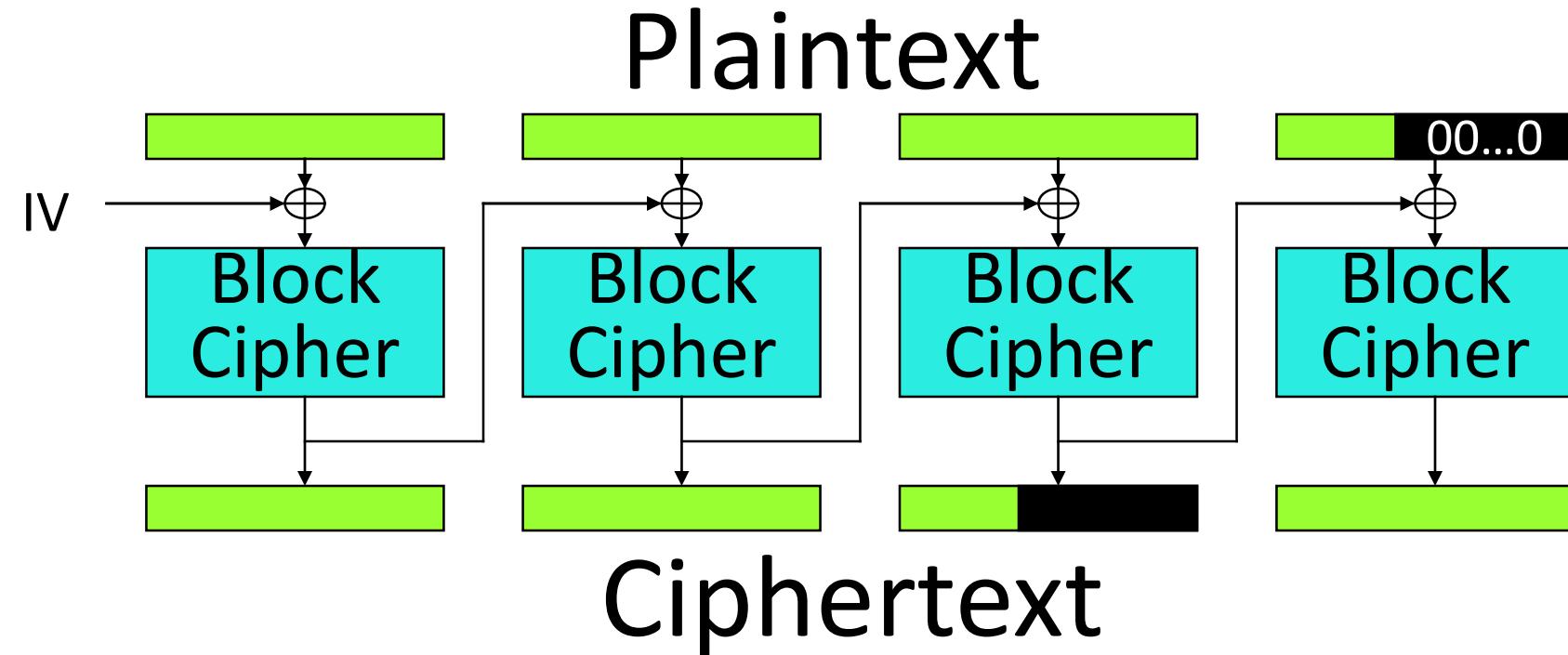


Some CBC-Mode Myths

- I can't use CBC mode because I need random-access decryption.
- I can't use a block cipher because of data expansion.

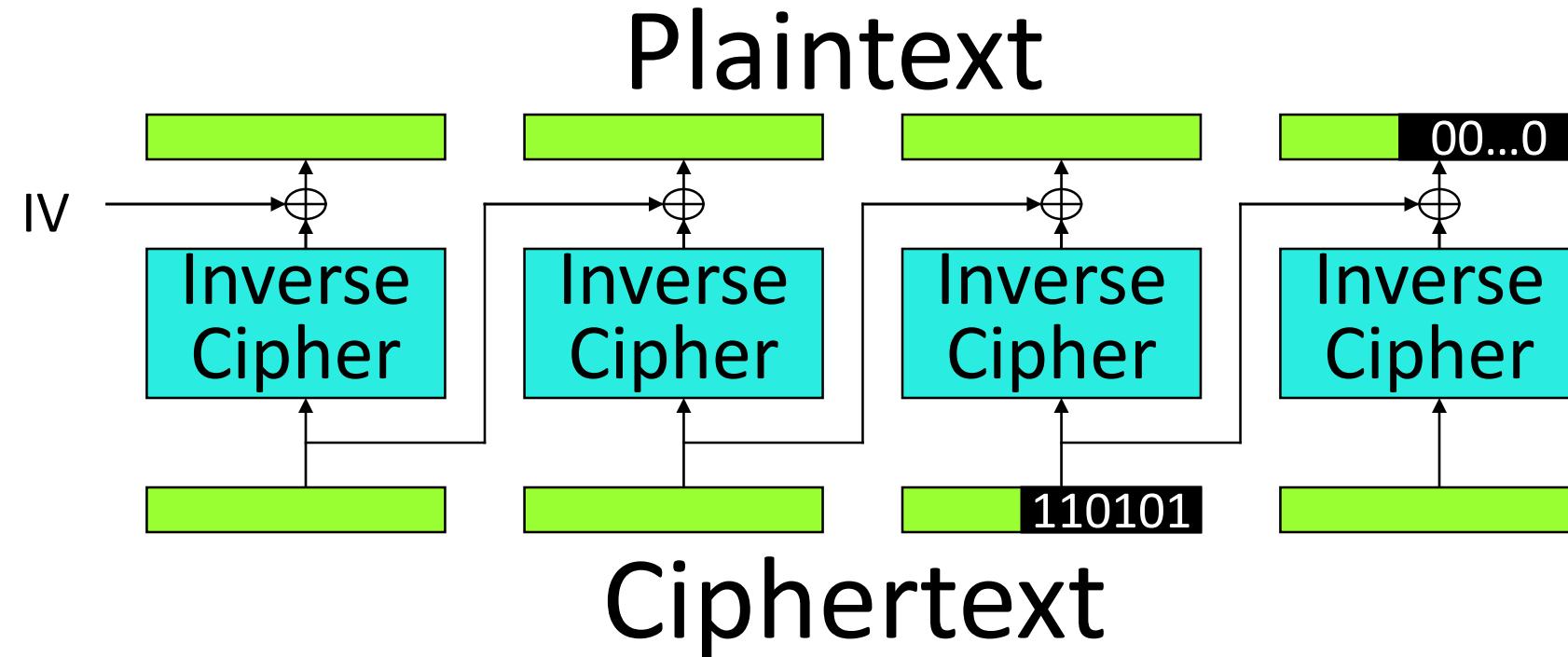
Ciphertext Stealing

Cipher Block Chaining (CBC) Encryption:



Ciphertext Stealing

Cipher Block Chaining (CBC) Decryption:



Some CBC-Mode Myths

- I can't use CBC mode because I need random-access decryption.
- I can't use a block cipher because of data expansion.

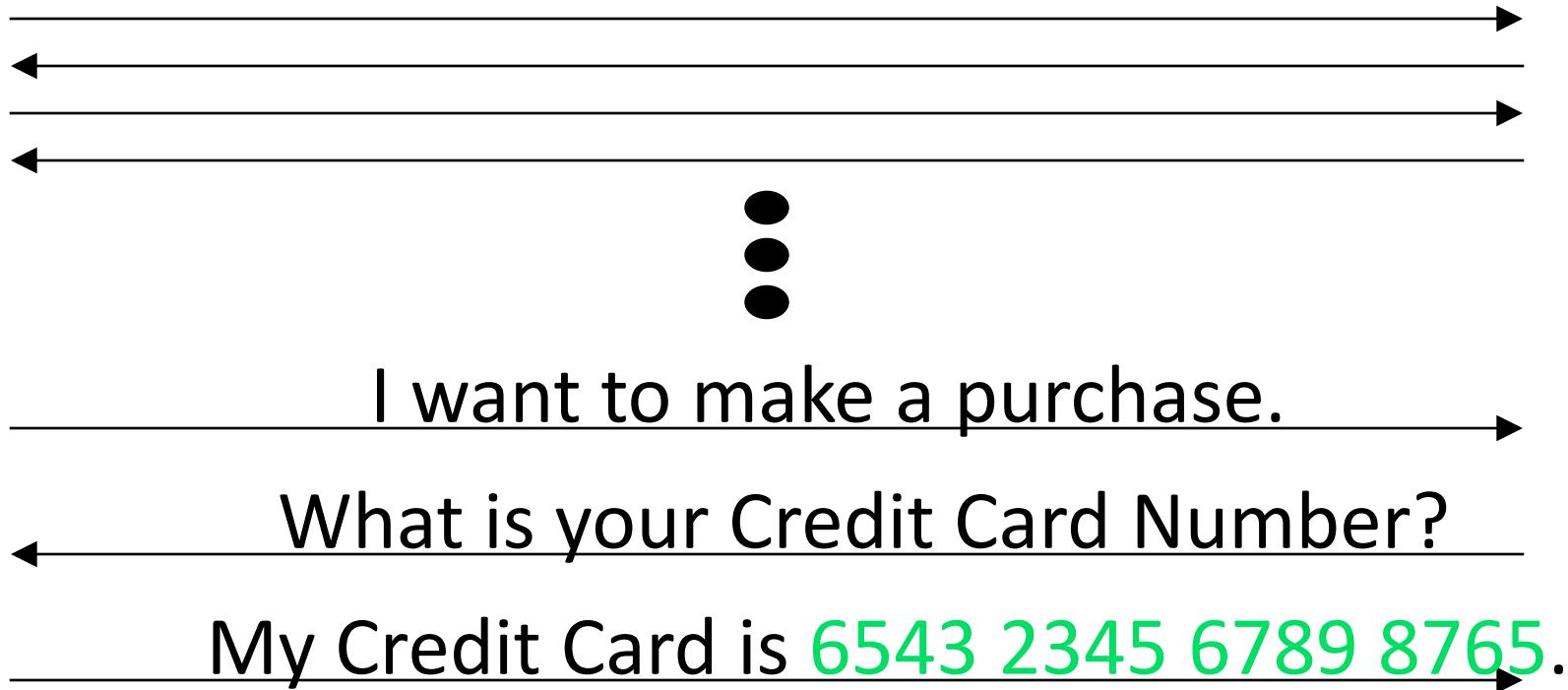
Some CBC-Mode Myths

- I can't use CBC mode because I need random-access decryption.
- I can't use a block cipher because of data expansion.

Transfer of Confidential Data

You (client)

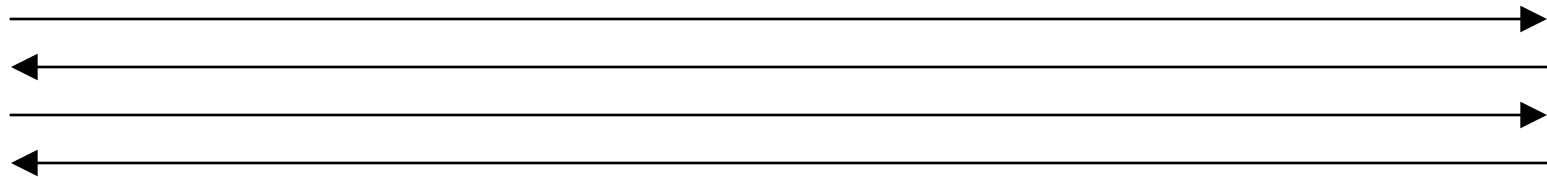
Merchant (server)



Transfer of Confidential Data

You (client)

Merchant (server)



I want to make a purchase.



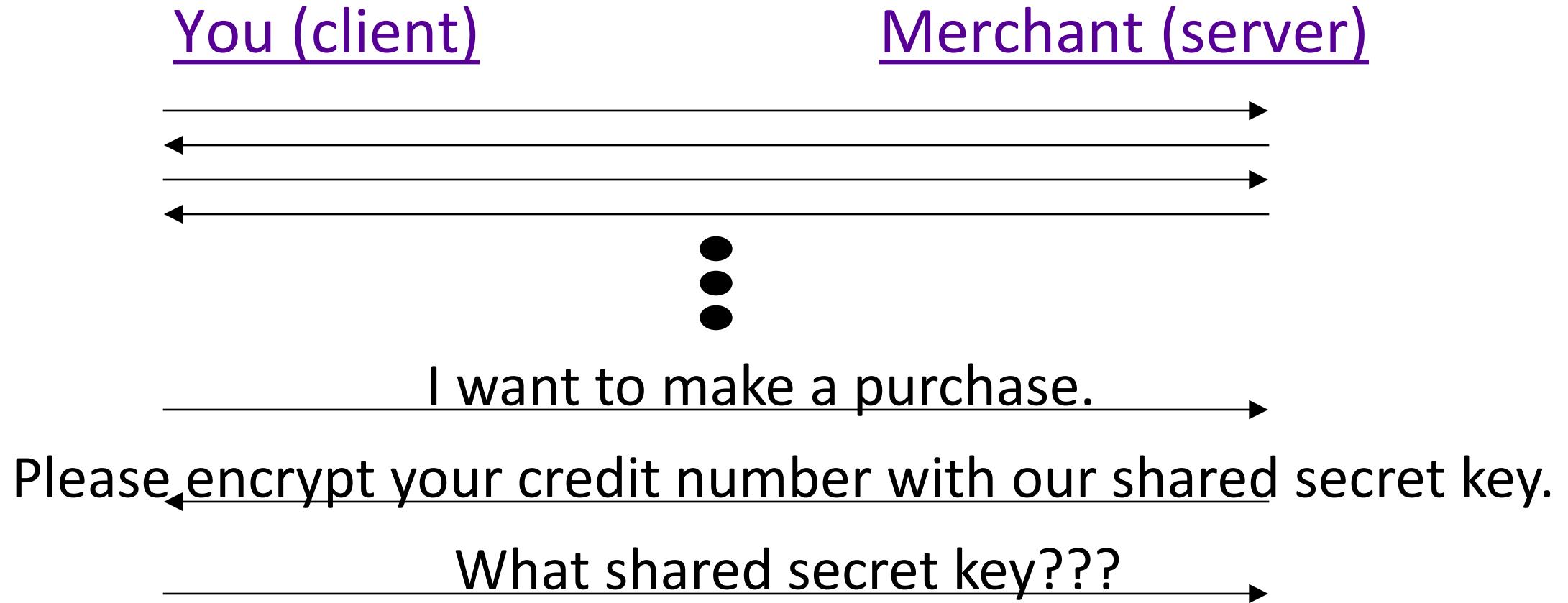
What is your Credit Card Number?



My Credit Card is *E(6543 2345 6789 8765)*.



Transfer of Confidential Data



Asymmetric Encryption

- What if the user and merchant have no prior relationship?
- Asymmetric encryption allows someone to encrypt a message for a recipient without knowledge of the recipient's decryption key.

The Fundamental Equation

$$Z \equiv Y^X \pmod{N}$$

The Fundamental Equation

$$Z \equiv Y^X \pmod{N}$$

When Z is unknown, it can be efficiently computed.

The Fundamental Equation

$$Z \equiv Y^X \pmod{N}$$

When X is unknown, the problem is known as the *discrete logarithm* and is generally believed to be hard to solve.

The Fundamental Equation

$$Z \equiv Y^X \pmod{N}$$

When Y is unknown, the problem is known as *discrete root finding* and is generally believed to be hard to solve ... without the factorization of N .

The Fundamental Equation

$$Z \equiv Y^X \pmod{N}$$

The problem is not well-studied for the case when N is unknown.

How to compute $Y^X \bmod N$

Compute Y^X and then reduce $\bmod N$.

- If X , Y , and N each are 2,048-bit integers, Y^X consists of $\sim 2^{2059}$ bits.
- Since there are roughly 2^{250} particles in the universe, storage is a problem.

How to compute $Y^X \bmod N$

- Repeatedly multiplying by Y by itself X times (with a modulo N reduction after each multiplication) solves the storage problem.
- However, we would need to perform $\sim 2^{1900}$ 64-bit multiplications per second to complete the computation before the sun burns out.

How to compute $Y^X \bmod N$

Multiplication by Repeated Doubling

To compute $X \bullet Y$,

compute $Y, 2Y, 4Y, 8Y, 16Y, \dots$

and sum up those values dictated by the binary representation of X .

How to compute $Y^X \bmod N$

Multiplication by Repeated Doubling

To compute $X \bullet Y$,

compute $Y, [2Y], 4Y, [8Y], [16Y], \dots$

and sum up those values dictated by the binary representation of X .

Example: $26Y = 2Y + 8Y + 16Y.$

How to compute $Y^X \bmod N$

Exponentiation by Repeated Squaring

To compute Y^X ,

compute $Y, [Y^2], Y^4, [Y^8], [Y^{16}], \dots$

and multiply those values dictated by the binary representation of X .

Example: $Y^{26} = Y^2 \bullet Y^8 \bullet Y^{16}$.

How to compute $Y^X \bmod N$

We can now perform a 2,000-bit modular exponentiation using $\sim 3,000$ 2,000-bit modular multiplications.

- 2,000 squarings: $Y, Y^2, Y^4, \dots, Y^{2^{2000}}$
- $\sim 1,000$ “ordinary” multiplications

The Fundamental Equation

$$Z \equiv Y^X \pmod{N}$$

When X is unknown, the problem is known as the *discrete logarithm* and is generally believed to be hard to solve.

Diffie-Hellman Key Exchange

Diffie-Hellman Key Exchange

Alice

Bob

Diffie-Hellman Key Exchange

Alice

- Randomly select a .

Bob

- Randomly select b .

Diffie-Hellman Key Exchange

Alice

- Randomly select a .
- Send $A = Y^a \pmod{N}$.

Bob

- Randomly select b .
- Send $B = Y^b \pmod{N}$.

Diffie-Hellman Key Exchange

Alice

- Randomly select a .
- Send $A = Y^a \pmod{N}$.
- Compute $K = B^a \pmod{N}$.

Bob

- Randomly select b .
- Send $B = Y^b \pmod{N}$.
- Compute $K = A^b \pmod{N}$.

Diffie-Hellman Key Exchange

Alice

- Randomly select a .
- Send $A = Y^a \pmod{N}$.
- Compute $K = B^a \pmod{N}$.

Bob

- Randomly select b .
- Send $B = Y^b \pmod{N}$.
- Compute $K = A^b \pmod{N}$.

$$B^a = (Y^b)^a = Y^{ba} = Y^{ab} = (Y^a)^b = A^b$$

The Fundamental Equation

$$Z \equiv Y^X \pmod{N}$$

When Y is unknown, the problem is known as *discrete root finding* and is generally believed to be hard to solve ... without the factorization of N .

RSA Encryption/Decryption

- Select two large primes p and q .
- Publish the product $N = pq$.
- The exponent X is typically fixed at 65537.
- Encrypt message Y as $E(Y) = Y^X \bmod N$.
- Decrypt ciphertext Z as $D(Z) = Z^{1/X} \bmod N$.
- Note $D(E(Y)) = (Y^X)^{1/X} \bmod N = Y$.

How do you compute $Z^{1/X} \bmod N$?

$E(M) = M^e \bmod N$ with $N = p \times q$.

$D(E(M)) = (M^e)^{1/e} \bmod N = (M^e)^d \bmod N$
if $d \times e = 1 \bmod (p - 1)(q - 1)$.

Example

Choose primes p and q and encryption exponent e .

Compute modulus N , and decryption exponent d .

$$E(M) = M^e \bmod N, D(Z) = Z^d \bmod N, D(E(M)) = M.$$

$$p = 5, q = 11, N = p \times q = 5 \times 11 = 55.$$

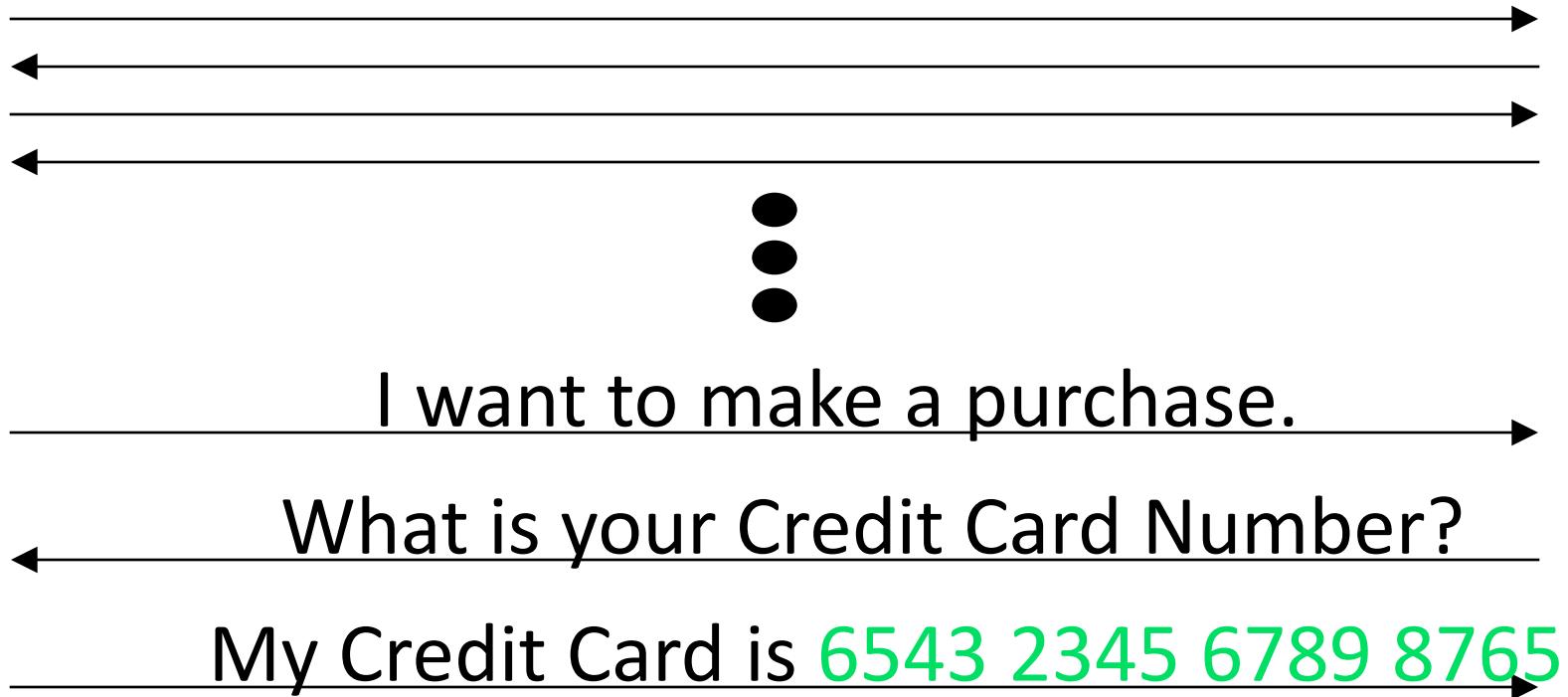
For $e = 7$, a suitable decryption exponent is $d = 23$, because

$$\begin{aligned} e \times d \bmod (p - 1)(q - 1) &= 7 \times 23 \bmod 4 \times 10 \\ &= 161 \bmod 40 = 1. \end{aligned}$$

Transfer of Confidential Data

You (client)

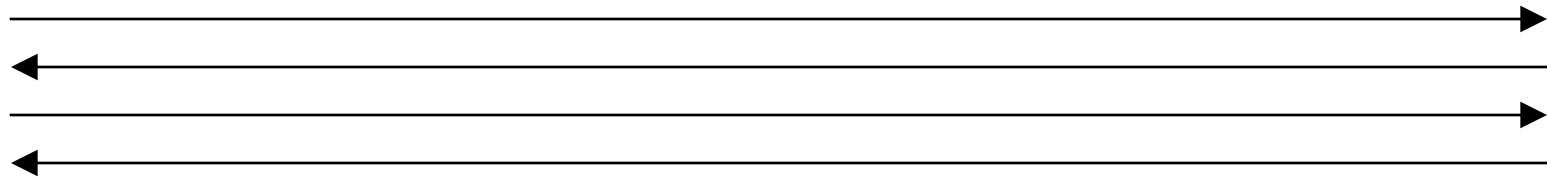
Merchant (server)



Transfer of Confidential Data

You (client)

Merchant (server)



I want to make a purchase.



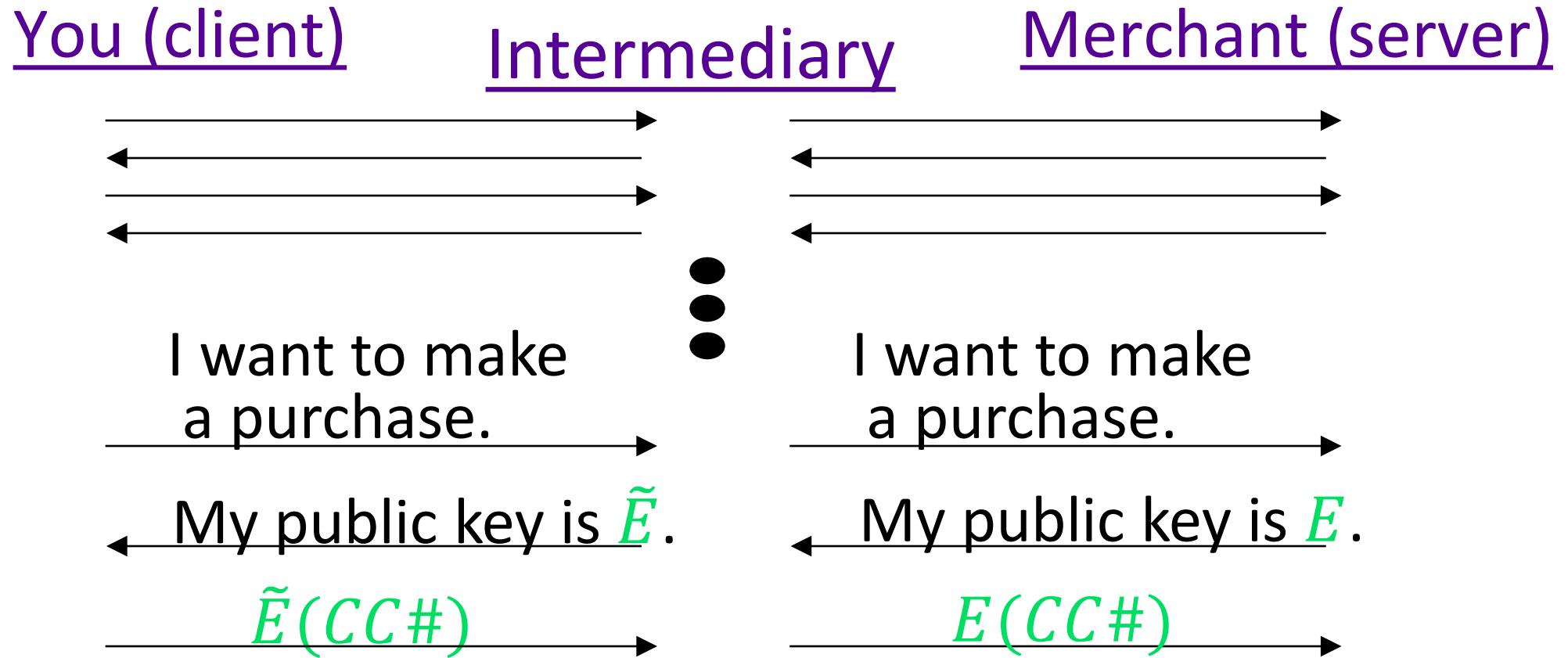
Here is my RSA public key E .



My Credit Card is $E(6543\ 2345\ 6789\ 8765)$.



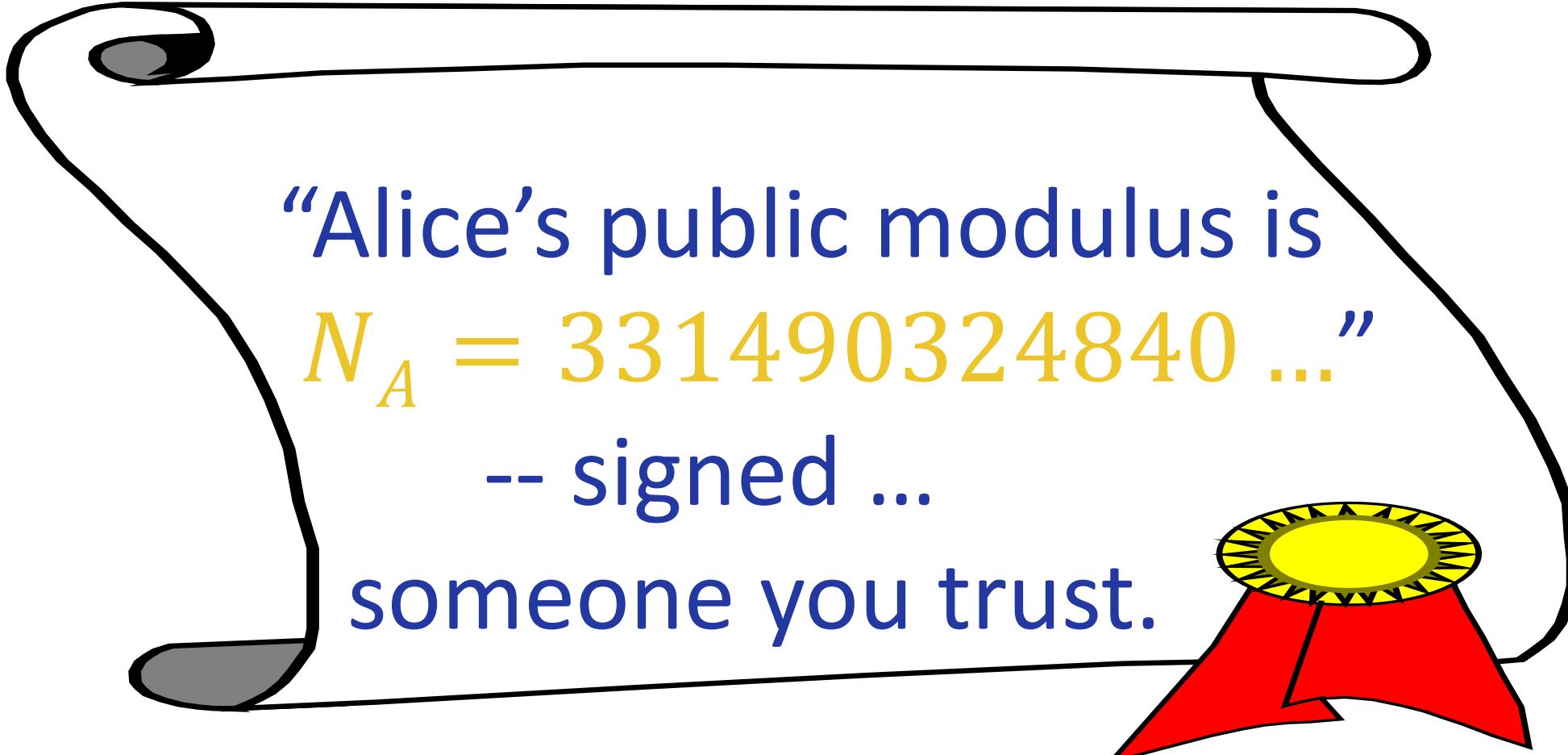
Intermediary Attack



RSA Signatures and Verification

- Not only is $D(E(Y)) = (Y^X)^{1/X} \bmod N = Y$, but also $E(D(Y)) = (Y^{1/X})^X \bmod N = Y$.
- To form a signature of message Y , create $S = D(Y) = Y^{1/X} \bmod N$.
- To verify the signature, check that $E(S) = S^X \bmod N$ matches Y .

Digital Certificates

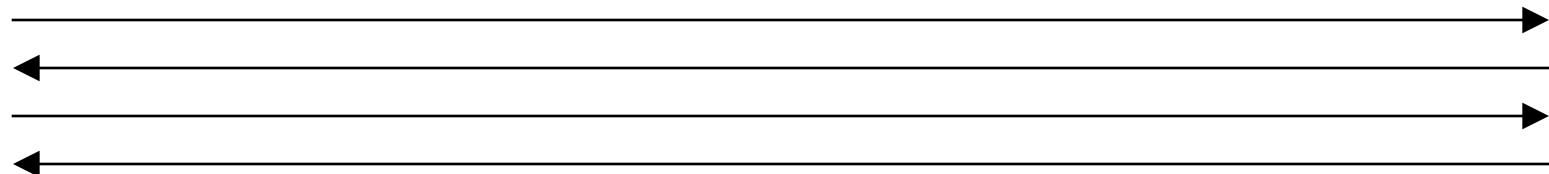


“Alice’s public modulus is
 $N_A = 331490324840 \dots$ ”
-- signed ...
someone you trust.

Transfer of Confidential Data

You (client)

Merchant (server)



I want to make a purchase.



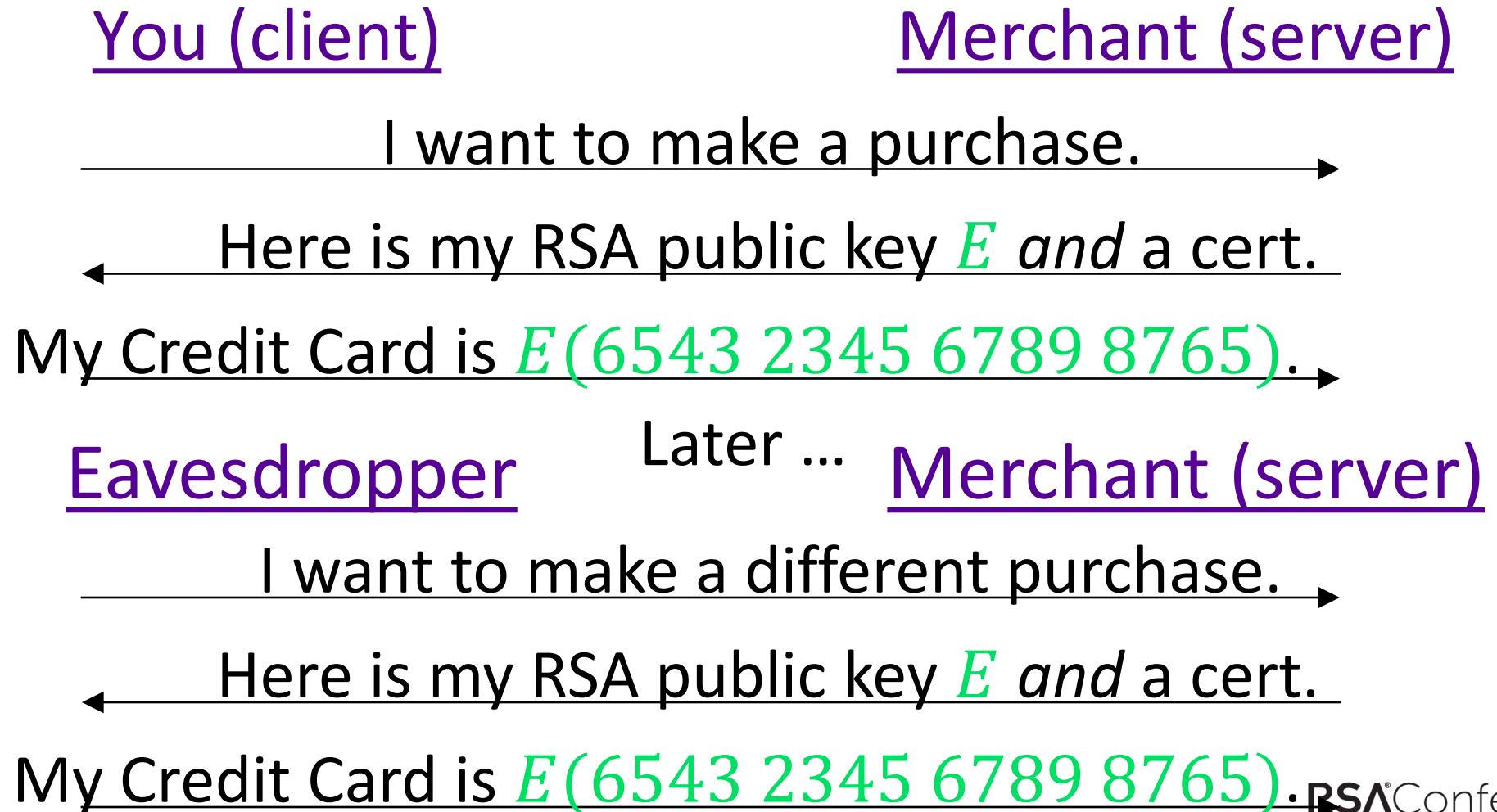
Here is my RSA public key E and a cert.



My Credit Card is $E(6543\ 2345\ 6789\ 8765)$.



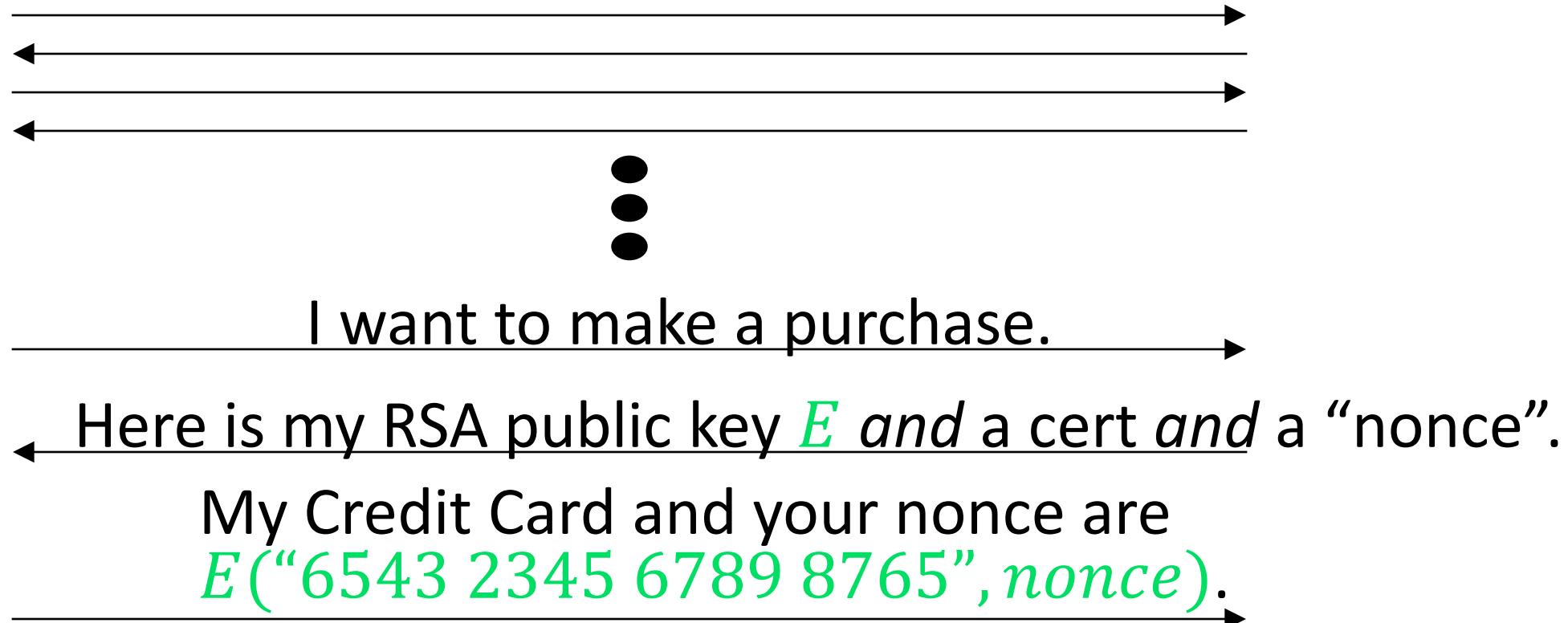
Replay Attack



Transfer of Confidential Data

You (client)

Merchant (server)



SSL/PCT/TLS History

- 1994: Secure Sockets Layer (SSL) v2.0
- 1995: Private Communication Technology (PCT) v1.0
- 1996: Secure Sockets Layer (SSL) v3.0
- 1997: Private Communication Technology (PCT) v4.0
- 1999: Transport Layer Security (TLS) v1.0
- 2006: Transport Layer Security (TLS) v1.1
- 2008: Transport Layer Security (TLS) v1.2
- 2018: Transport Layer Security (TLS) v1.3

SSL/PCT/TLS Handshake

You (client)

Merchant (server)

Let's talk securely.

Here are the protocols and ciphers I understand.

I choose this protocol and these ciphers.
Here is my public key, a cert, a nonce, etc.

Using your public key, I've encrypted a
random symmetric key (and your nonce).

SSL/PCT/TLS Secure Channel

Once the negotiation is complete, *all* subsequent secure messages are sent

- encrypted – using the negotiated session key, and
- integrity checked with a keyed “message authentication code”.

Hybrid Cryptography

- Asymmetric cryptography has many useful features not available in traditional symmetric cryptography.
- Symmetric cryptography is *much* more efficient than asymmetric.
- The practical hybrid is formed by using asymmetric cryptography to establish a secure channel and symmetric cryptography within the secure channel.

SSL/PCT/TLS Agility

A principal reason for the success of SSL/TLS is its agility.

- The handshake negotiates symmetric and asymmetric ciphers, the hash function, and even the protocol's own version.
- This has allowed the protocol to survive and expand while many underlying primitives have been discredited or lost favor.

Forward Secrecy

- If RSA is used to encrypt symmetric keys, then if your RSA key is *ever compromised*, an eavesdropper who has been quietly collecting your encrypted messages can suddenly “go back in time” and decrypt all of your prior traffic.
- If instead, a new Diffie-Hellman key is exchanged with each new transmission and destroyed when the transmission is complete, attacks must be active and real-time and cannot decrypt prior traffic.

Elliptic Curve Cryptography

Elliptic Curve Cryptosystems

Just what are *elliptic curves*?

*They really don't have anything to do with ellipses.

Wikipedia

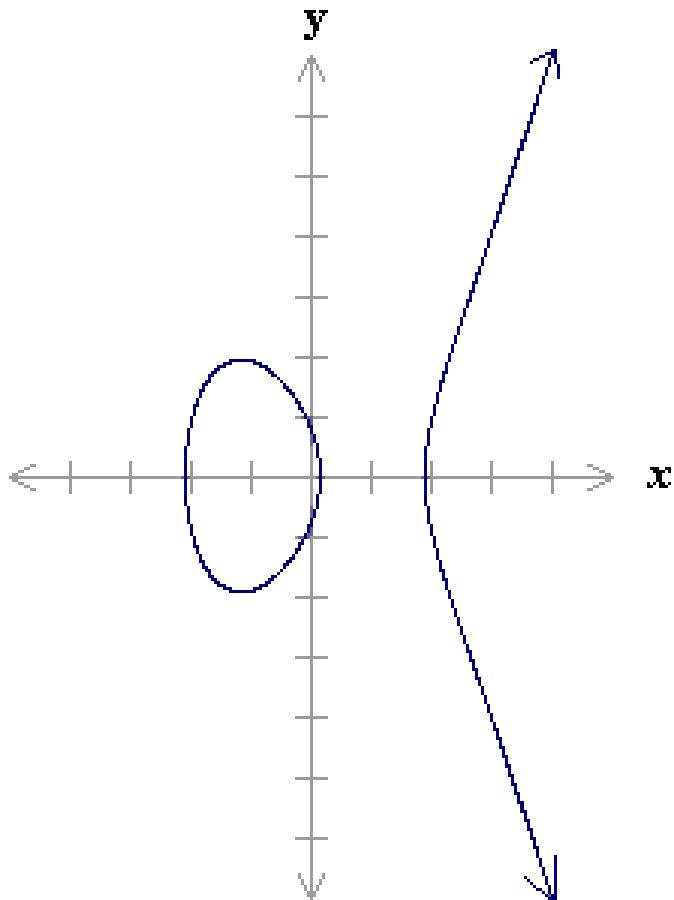
“In mathematics, an **elliptic curve (EC)** is a smooth, projective algebraic curve of genus one, on which there is a specified point O . An elliptic curve is in fact an abelian variety – that is, it has a multiplication defined algebraically, with respect to which it is a (necessarily commutative) group – and O serves as the identity element. Often the curve itself, without O specified, is called an elliptic curve.”

More Directly ...

An elliptic curve

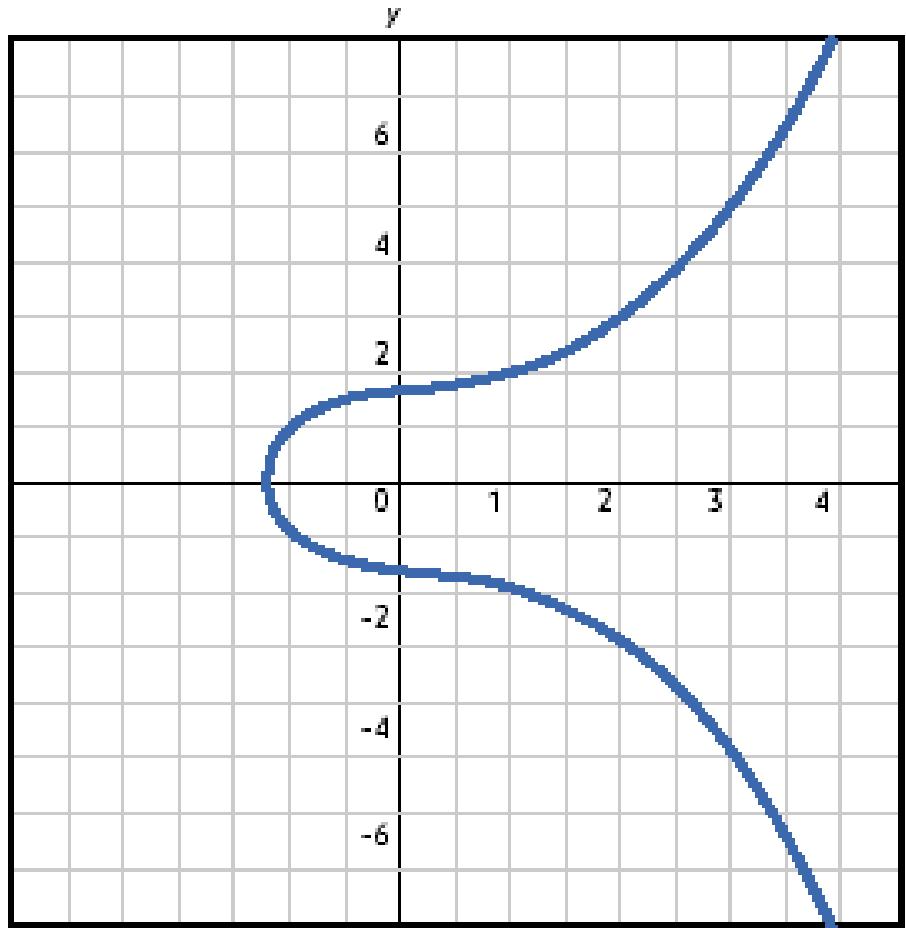
$$y^2 = x^3 + Ax + B$$

Some Elliptic Curves

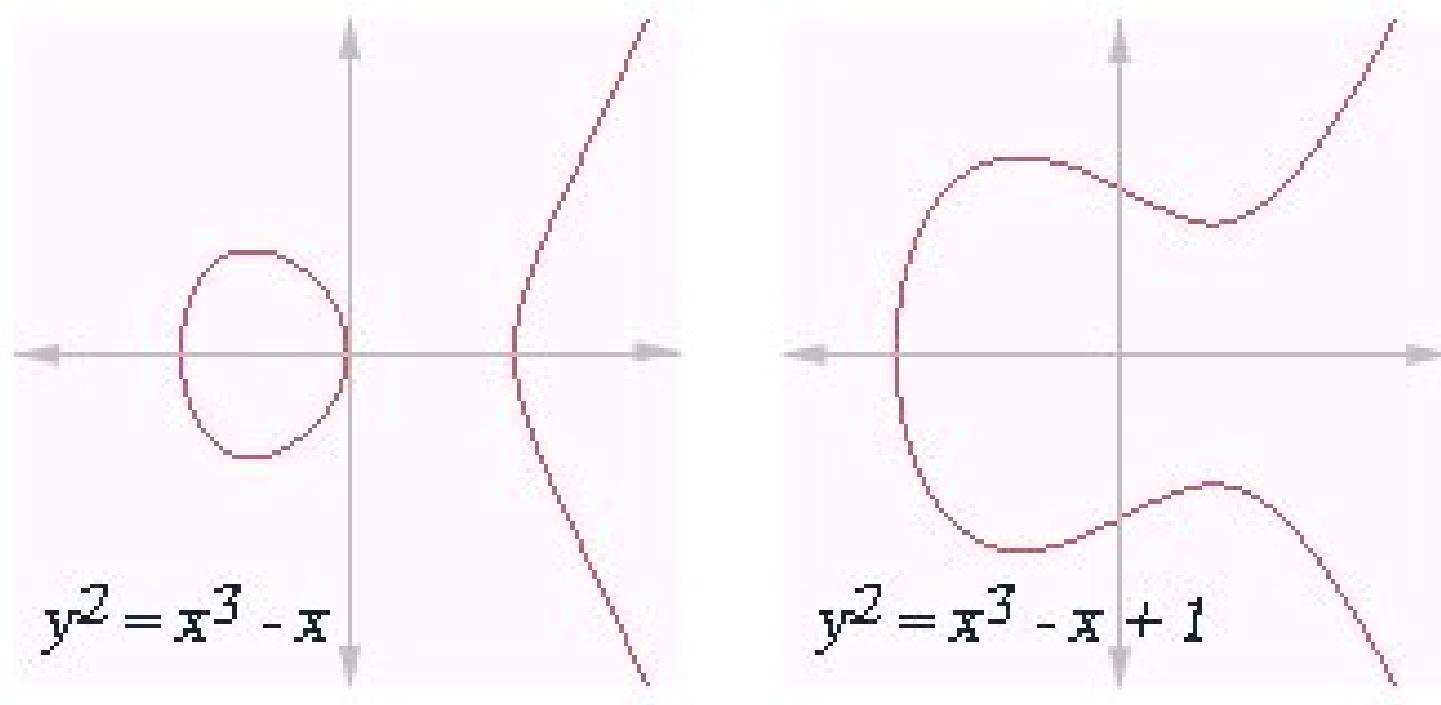


$$y^2 = x^3 - 4x + 0.67$$

Some Elliptic Curves

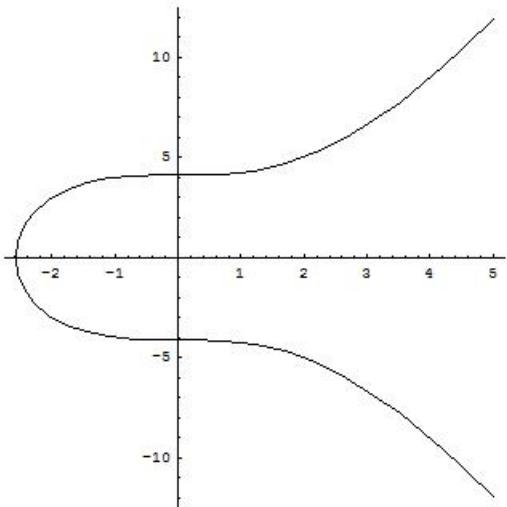


Some Elliptic Curves

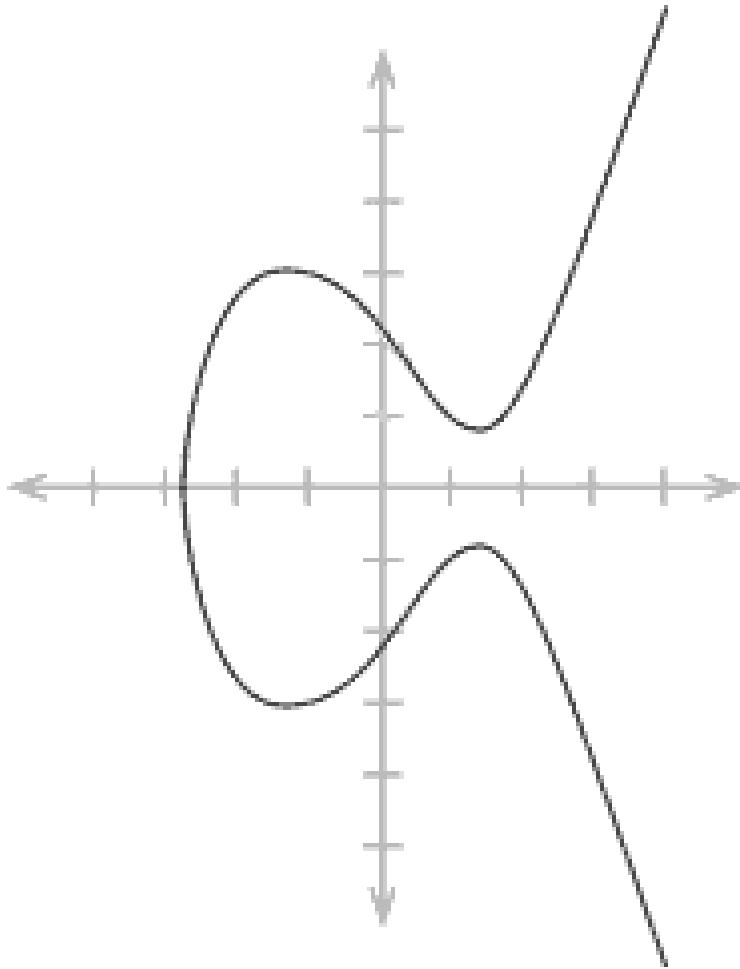


Some Elliptic Curves

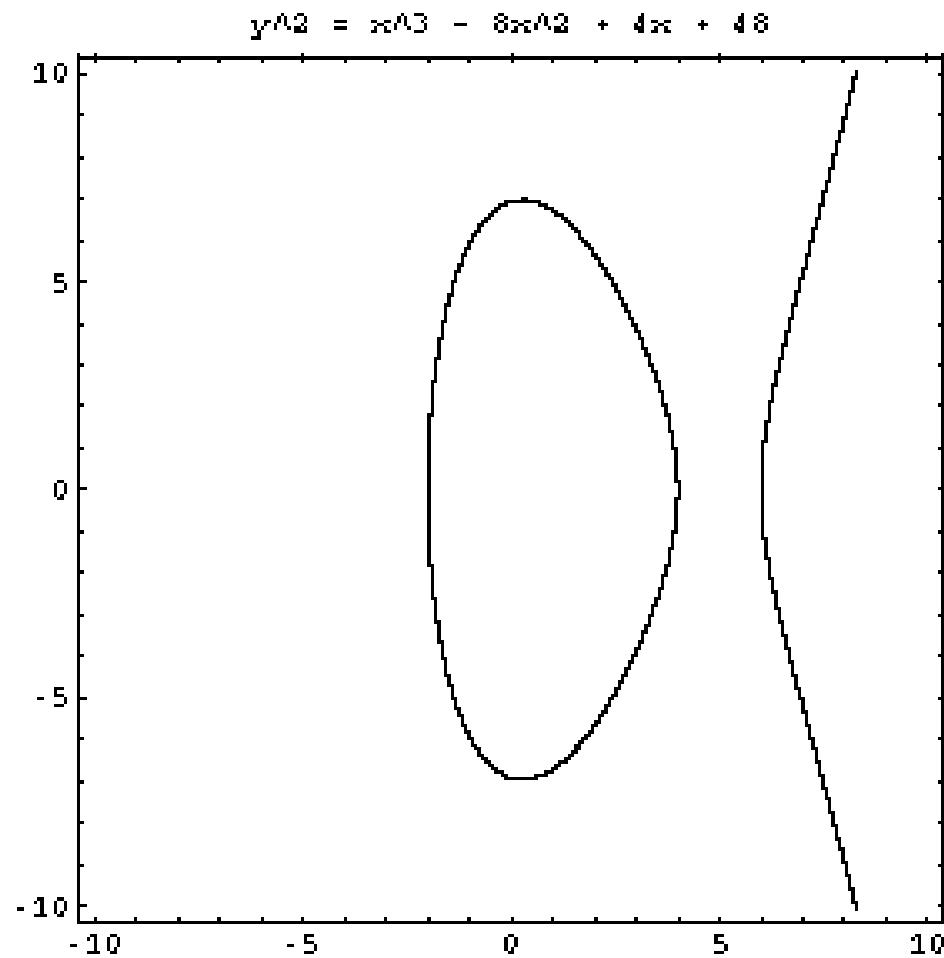
```
ImplicitPlot[y^2 - x^3 - 17 == 0, {x, -10, 5}, AspectRatio -> 1]
```



Some Elliptic Curves



Some Elliptic Curves



Elliptic Curves

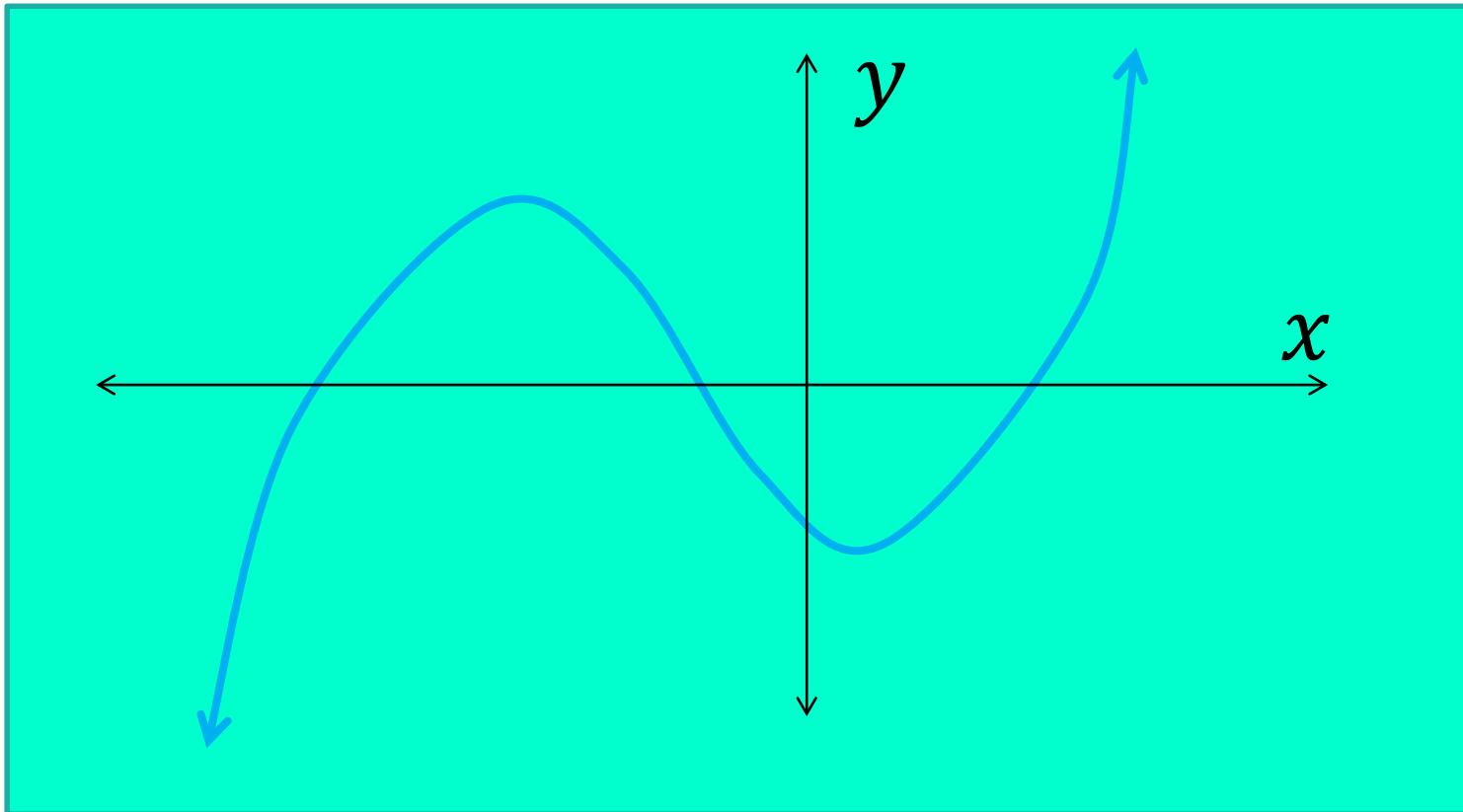
$$y^2 = x^3 + Ax + B$$

Elliptic Curves

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

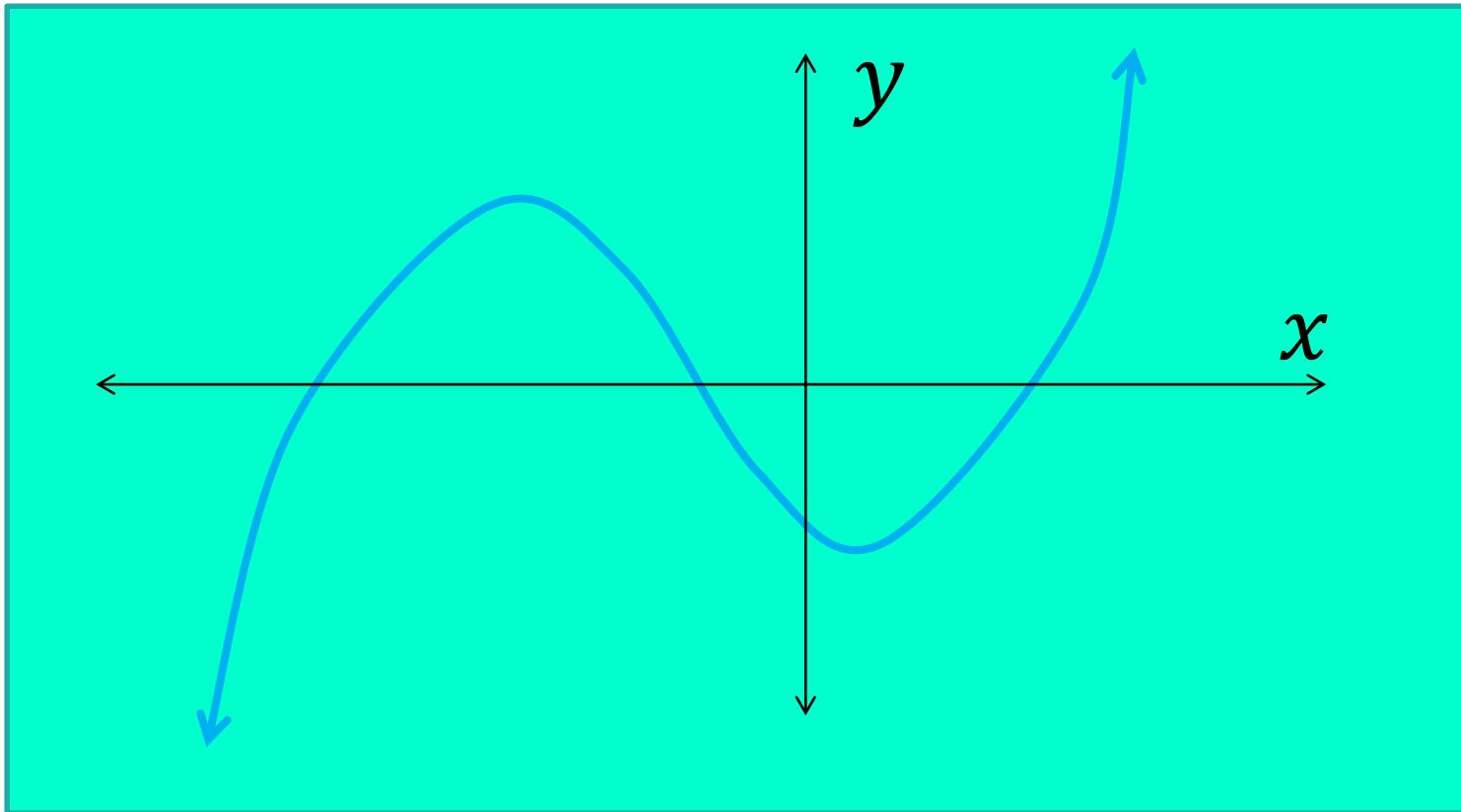
Elliptic Curves

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



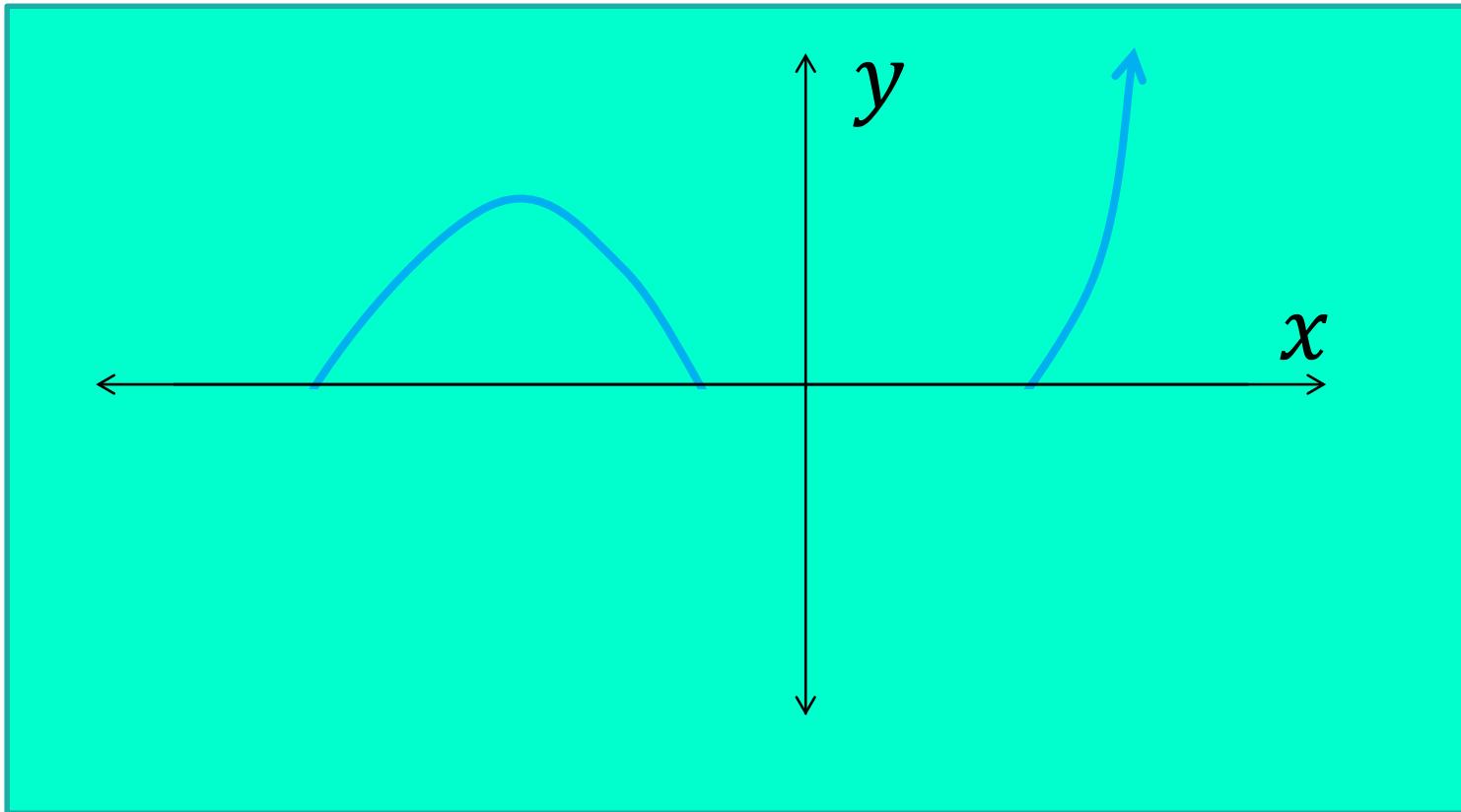
Elliptic Curves

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



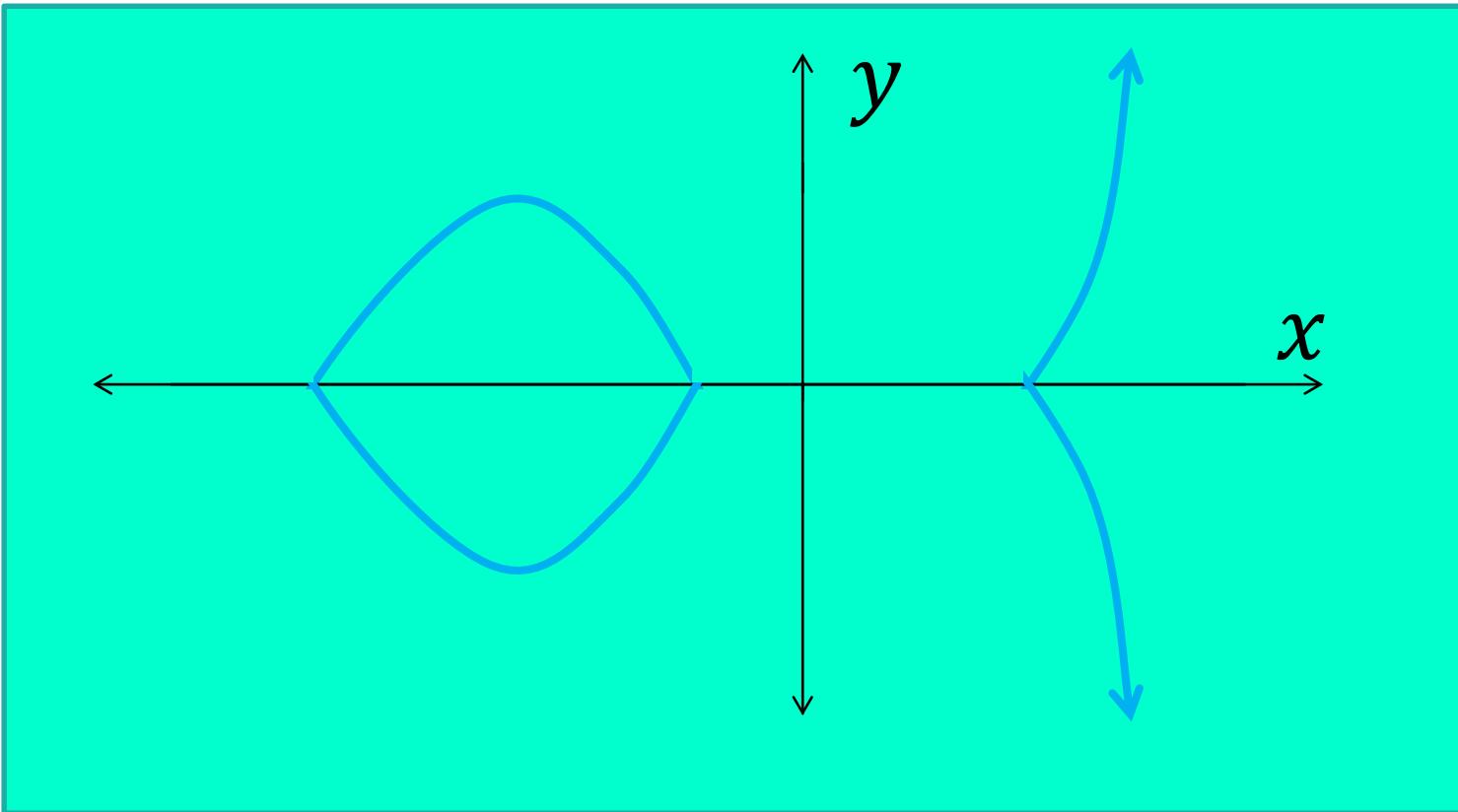
Elliptic Curves

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



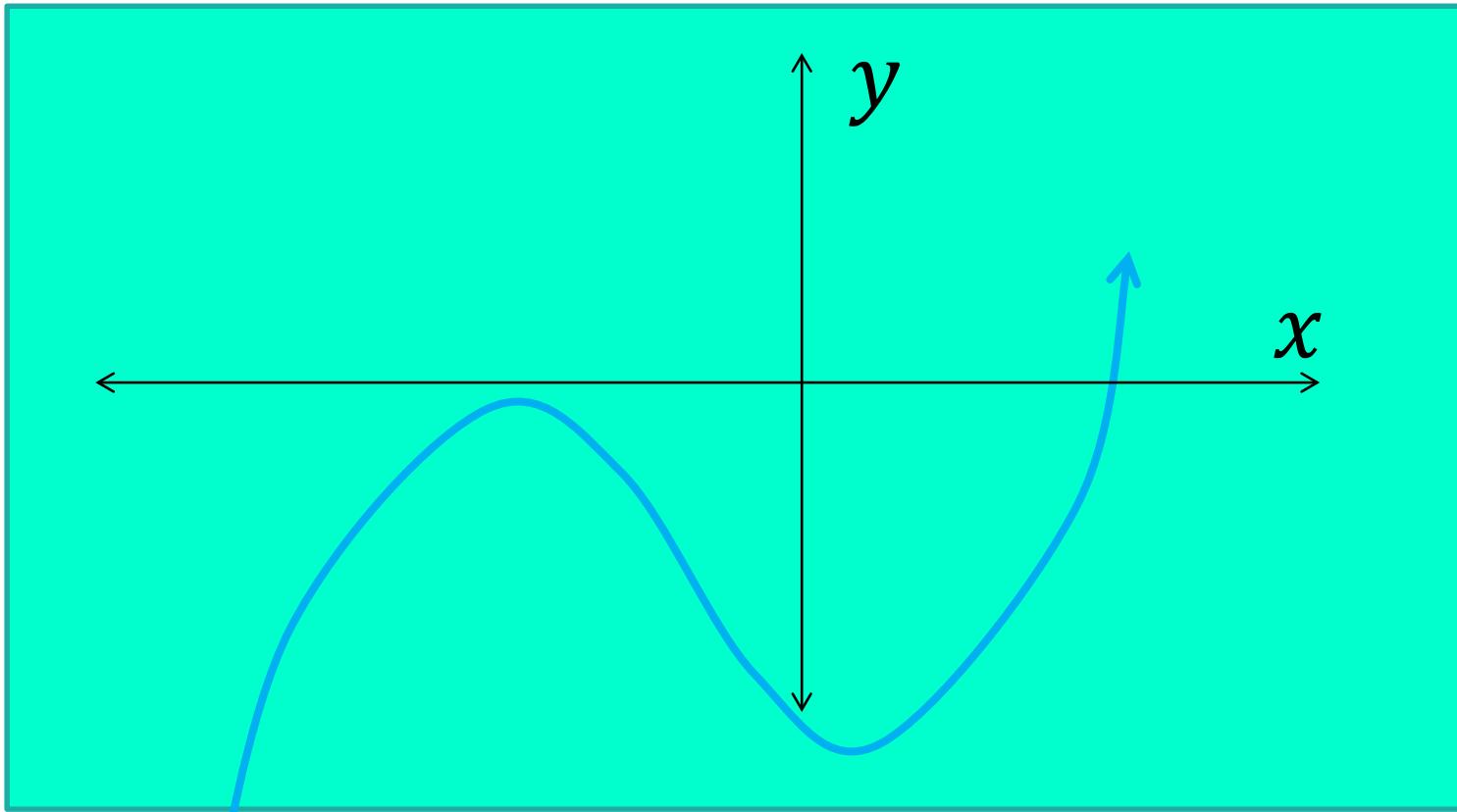
Elliptic Curves

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



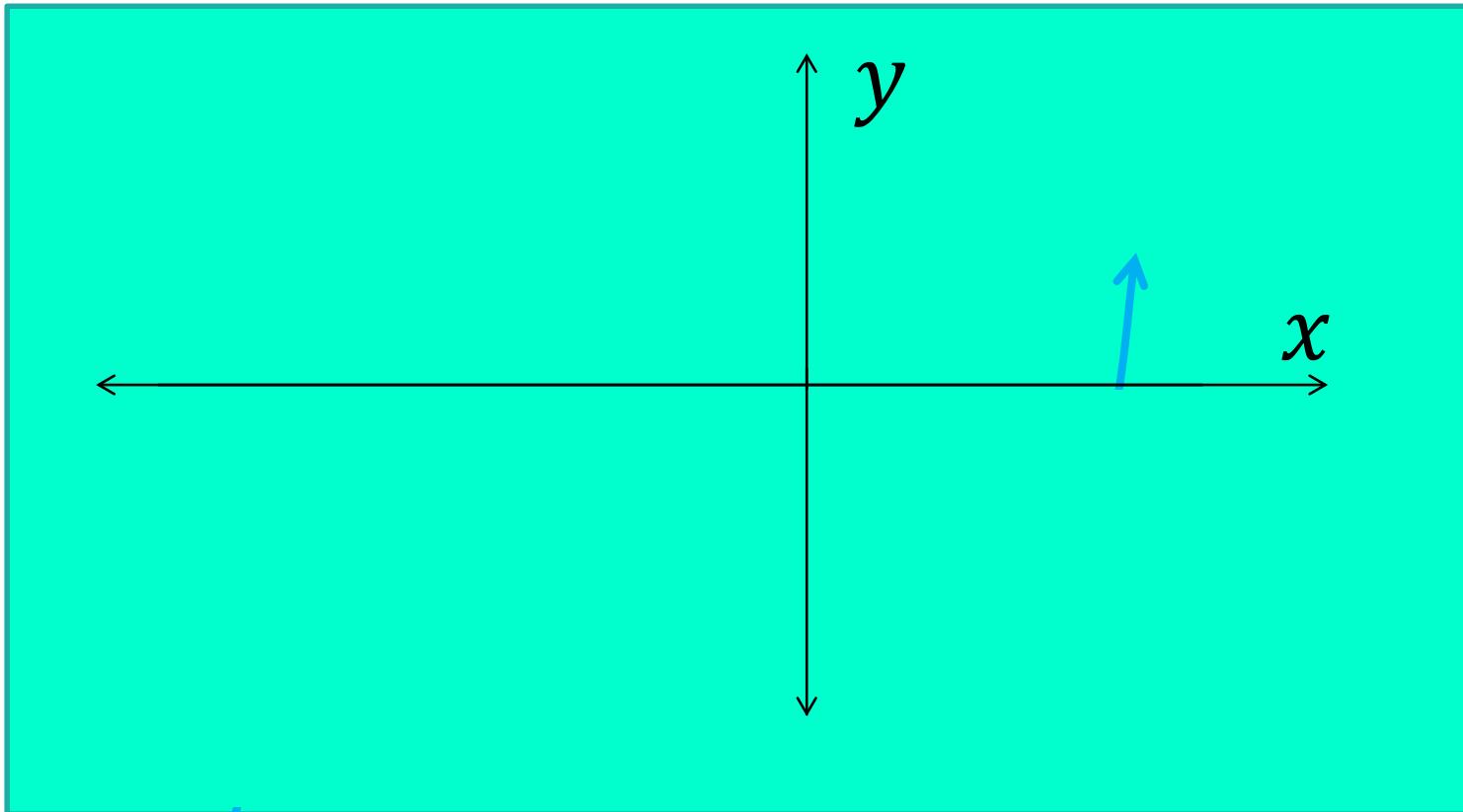
Elliptic Curves

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



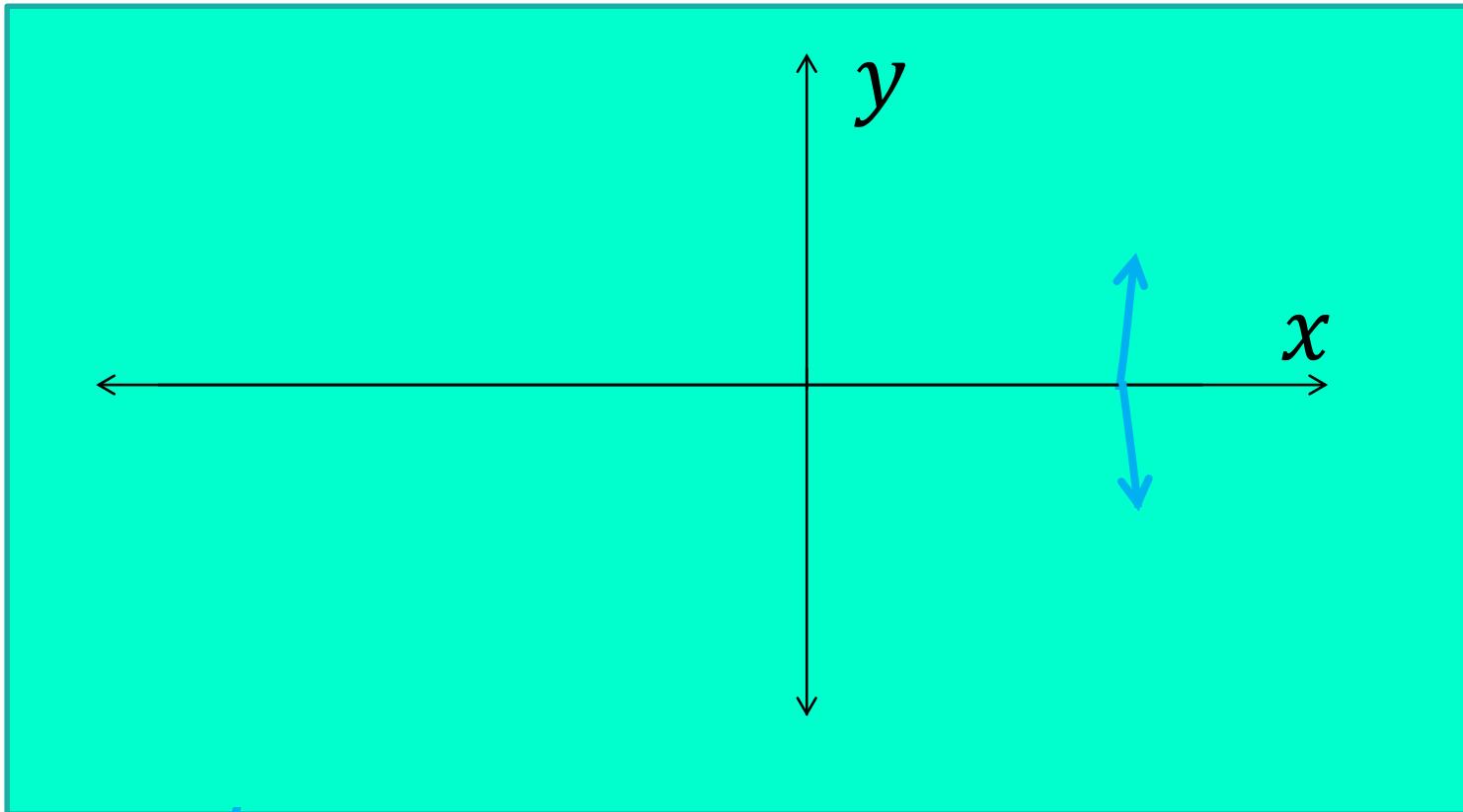
Elliptic Curves

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



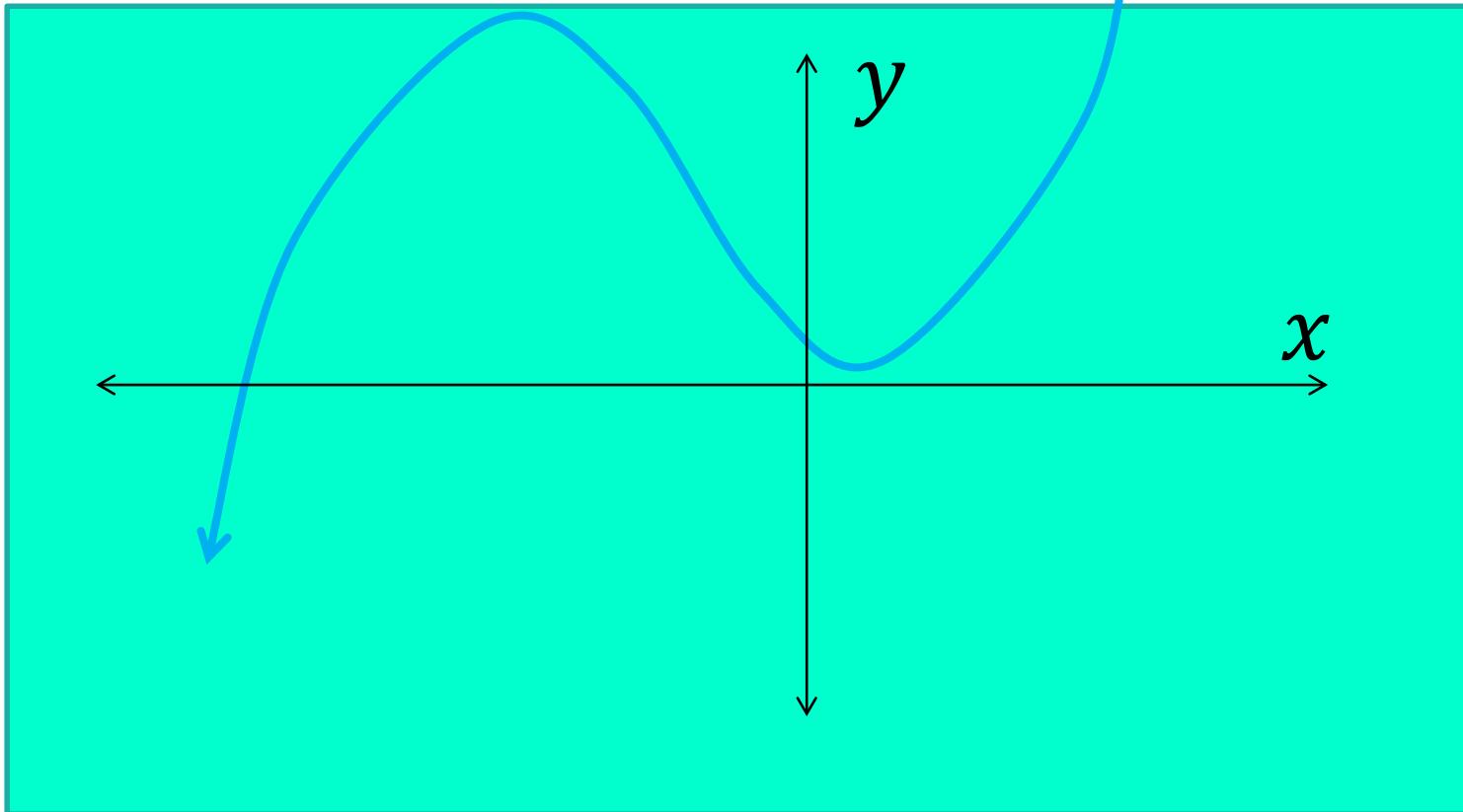
Elliptic Curves

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



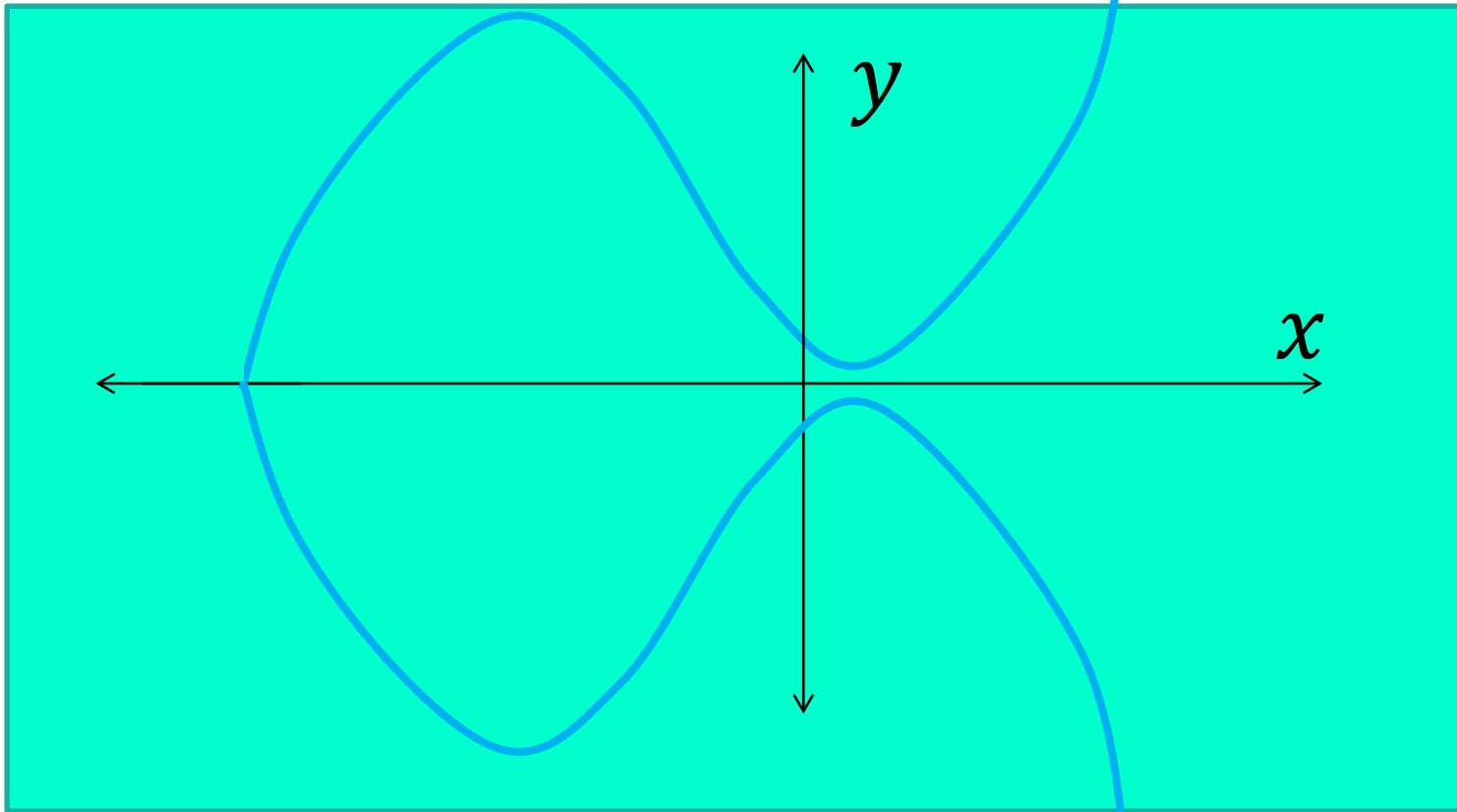
Elliptic Curves

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



Elliptic Curves

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



Double Roots

Things get weird when the local max or min just touches the x-axis, so we exclude this case by requiring that $4A^3 + 27B^2 \neq 0$.

Mathematical Groups

A mathematical *group* G is a set of objects together with a binary operation \times on those objects satisfying four properties.

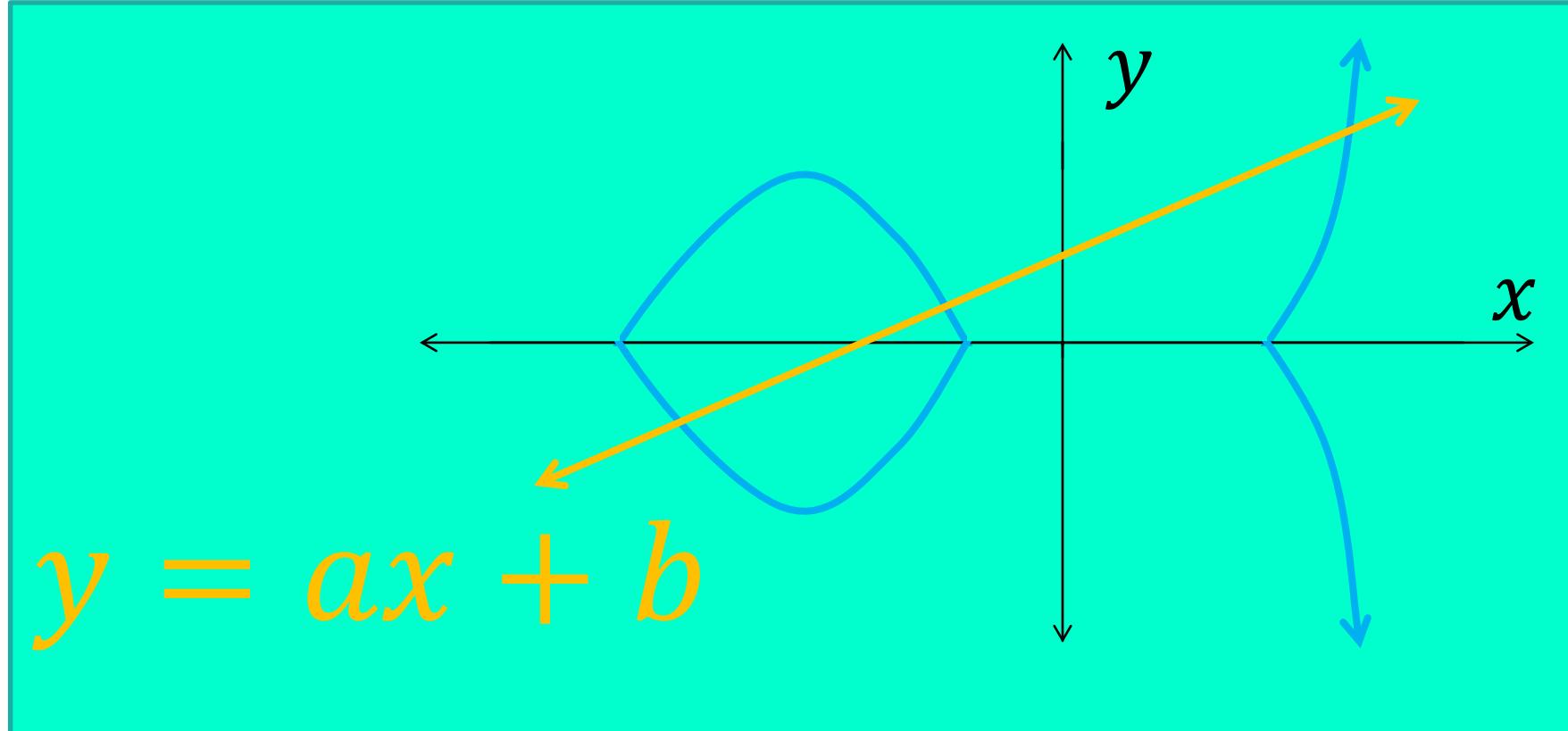
- Identity: $\mathbf{I} \in G$ such that for all $z \in G$, $z \times \mathbf{I} = z = \mathbf{I} \times z$.
- Inverses: For all $z \in G$, there exists $a \in G$, such that $a \times z = \mathbf{I} = z \times a$.
- Associativity: For all $a, b, c \in G$, $(a \times b) \times c = a \times (b \times c)$.
- Closure: For all $a, b \in G$, $a \times b \in G$.

Some Groups and Non-groups

- The integers (\mathbb{Z}) with addition (0 is the identity).
- The integers with subtraction, multiplication, or division.
- The rationals (\mathbb{Q}) with addition (0 is the identity).
- The rationals with subtraction, multiplication, or division.
- The non-zero rationals with multiplication (1 is the identity).
- $\mathbb{Z}_n = \{0, 1, 2, \dots, n - 1\}$ with modulo n addition (0 is the identity).
- $\mathbb{Z}_p^* = \{1, 2, \dots, p - 1\}$ with prime modulo p multiplication (1 is the identity).

Elliptic Curves Intersecting Lines

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



Elliptic Curves Intersecting Lines

Non-vertical Lines

Elliptic Curve: $y^2 = x^3 + Ax + B$

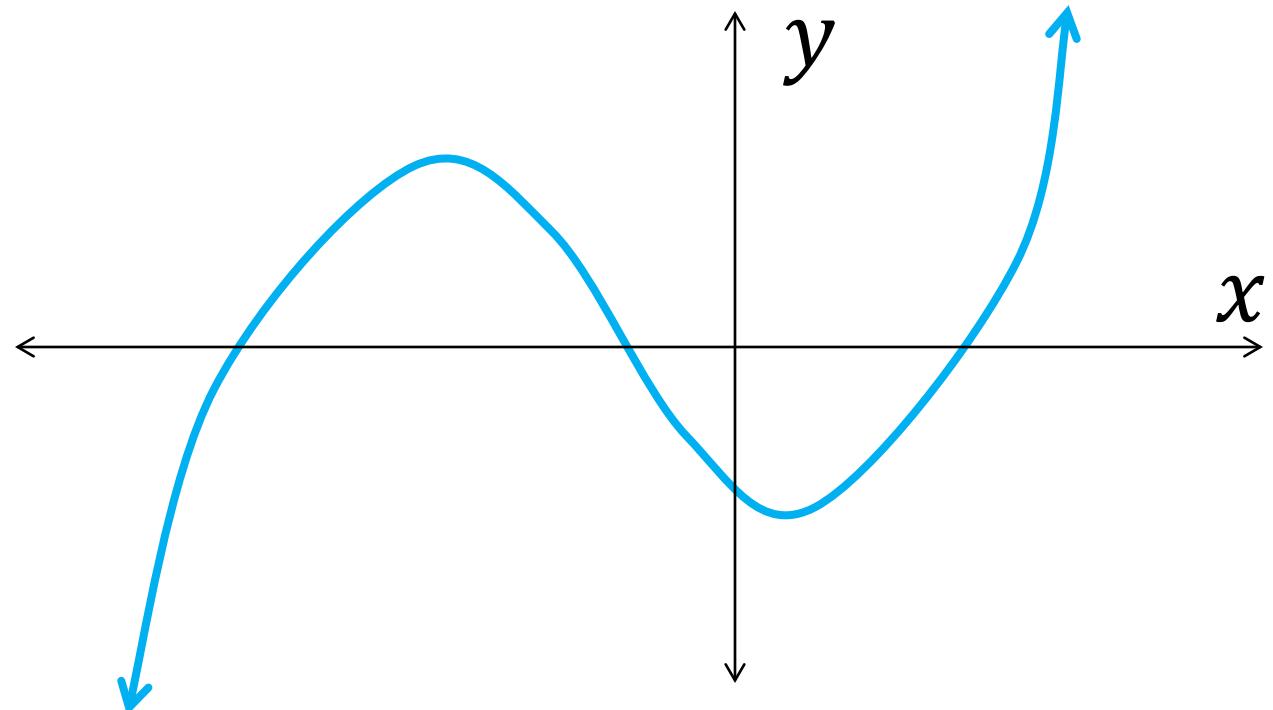
Straight Line: $y = ax + b$

$$(ax + b)^2 = x^3 + Ax + B$$

$$x^3 + A'x^2 + B'x + C' = 0$$

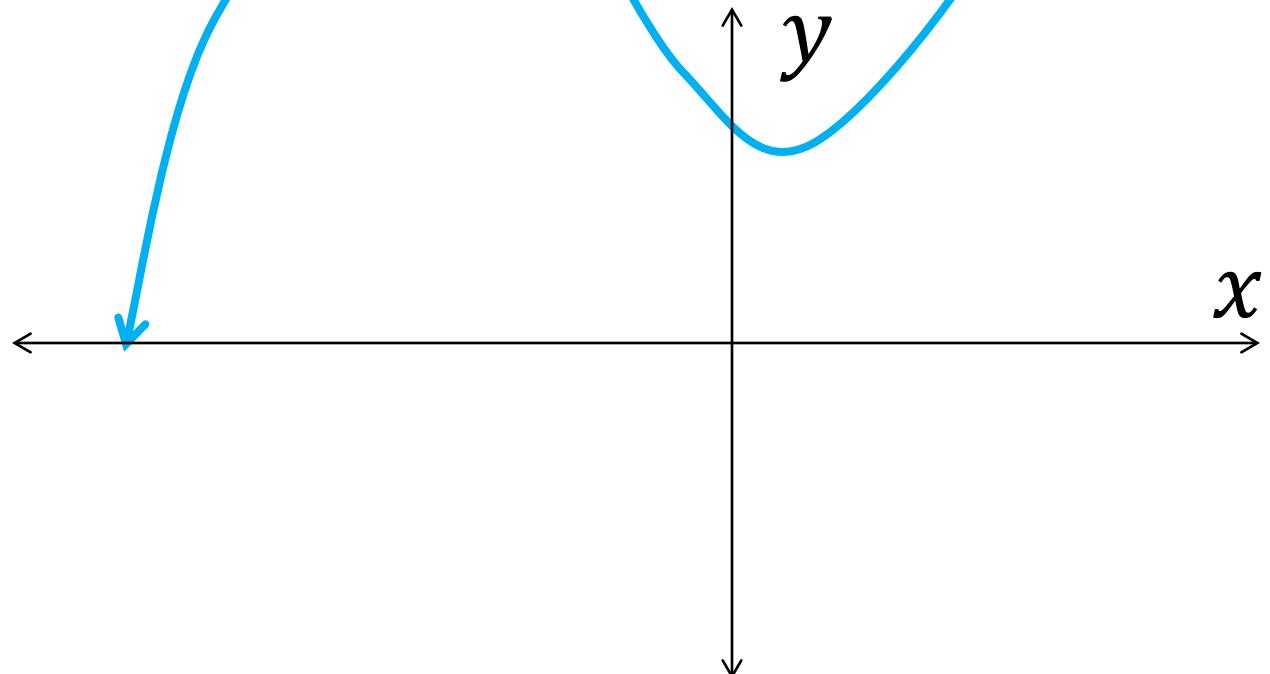
Elliptic Curves Intersecting Lines

$$x^3 + A'x^2 + B'x + C' = 0$$



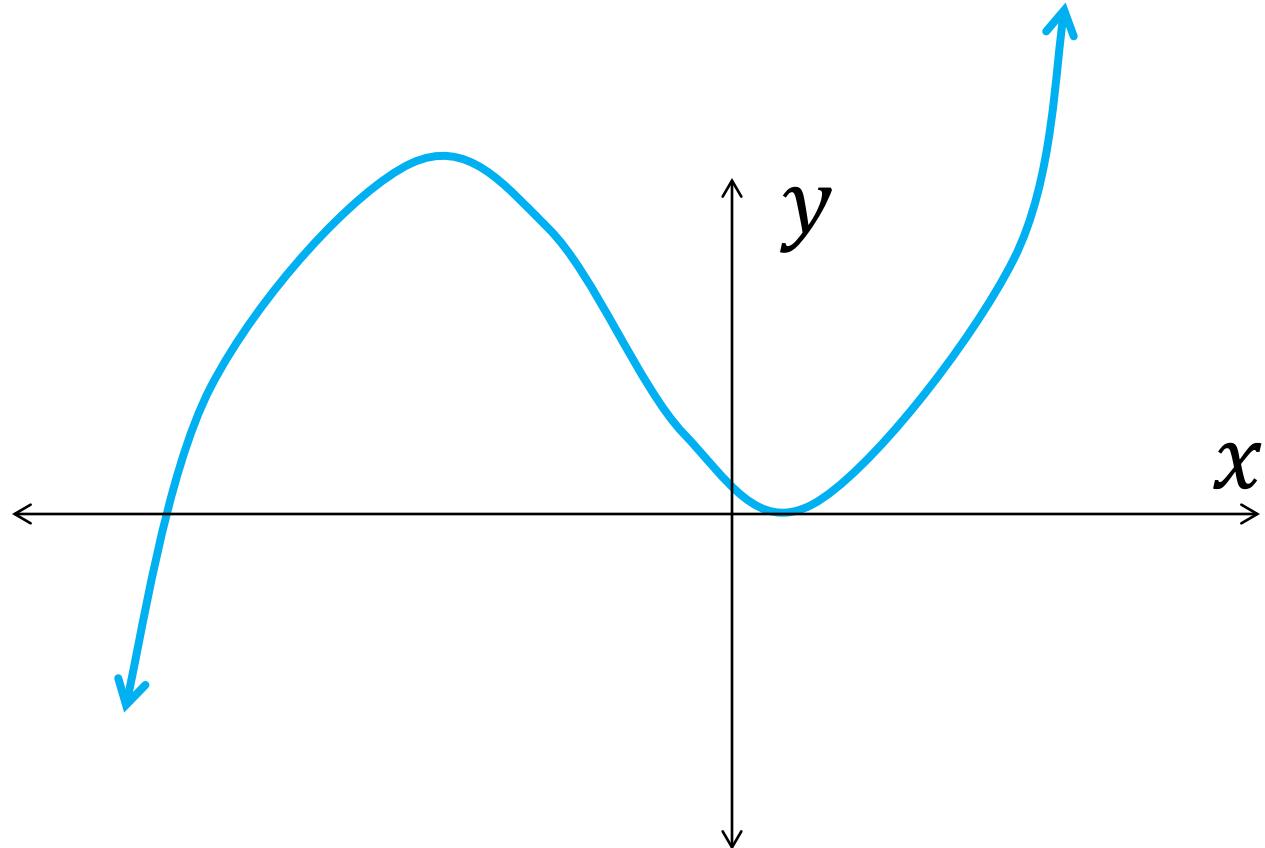
Elliptic Curves Intersecting Lines

$$x^3 + A'x^2 + B'x + C' = 0$$



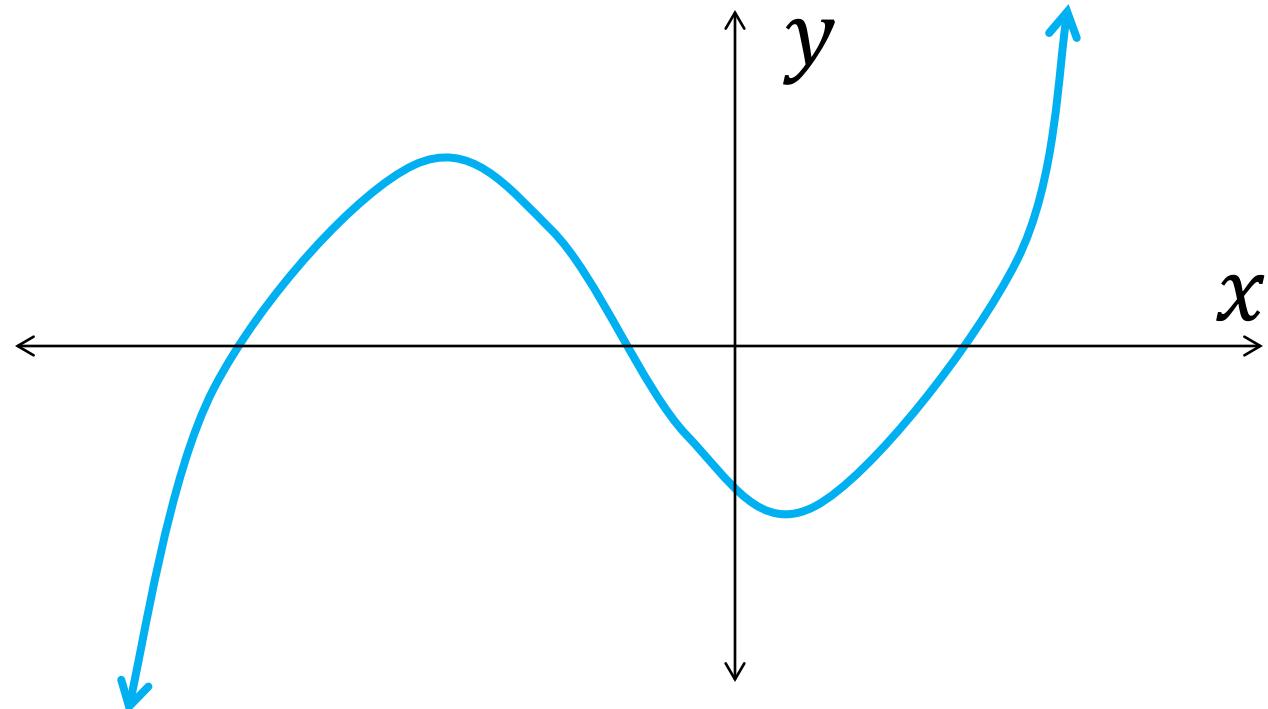
Elliptic Curves Intersecting Lines

$$x^3 + A'x^2 + B'x + C' = 0$$



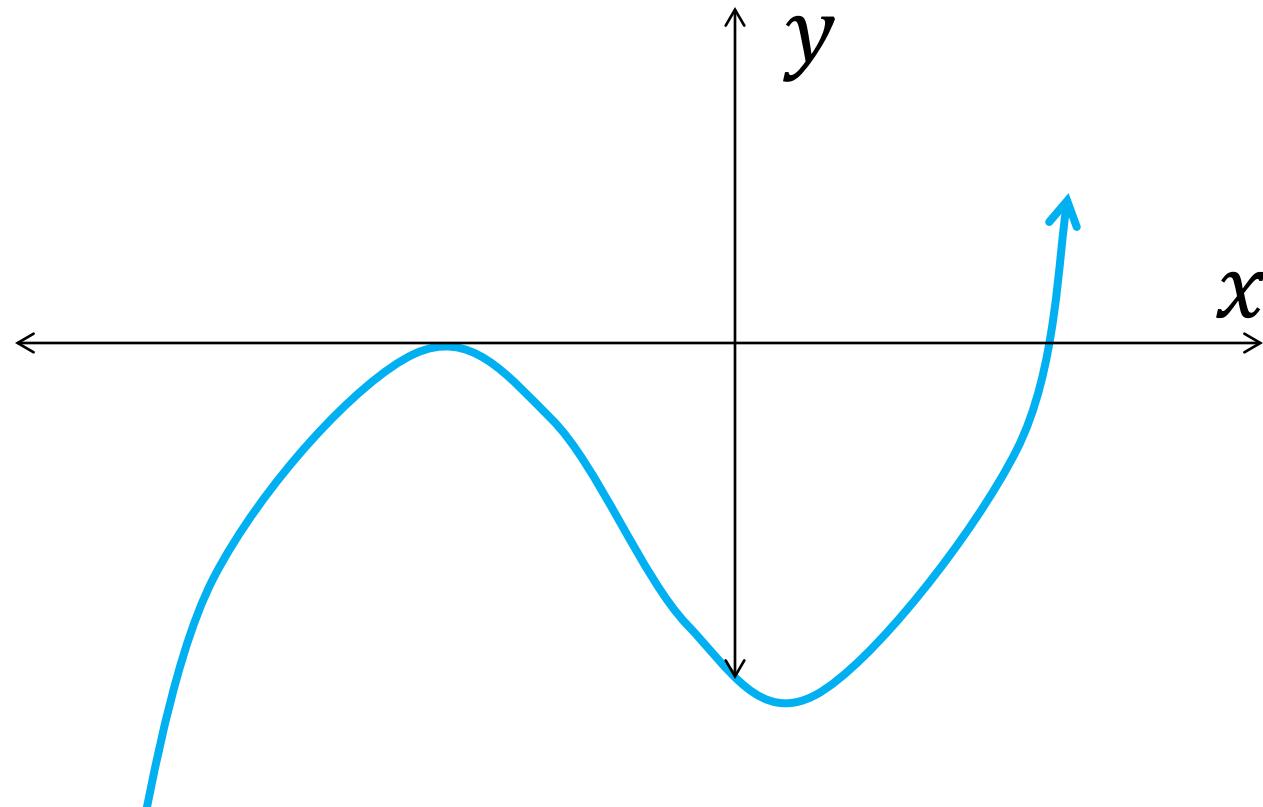
Elliptic Curves Intersecting Lines

$$x^3 + A'x^2 + B'x + C' = 0$$



Elliptic Curves Intersecting Lines

$$x^3 + A'x^2 + B'x + C' = 0$$



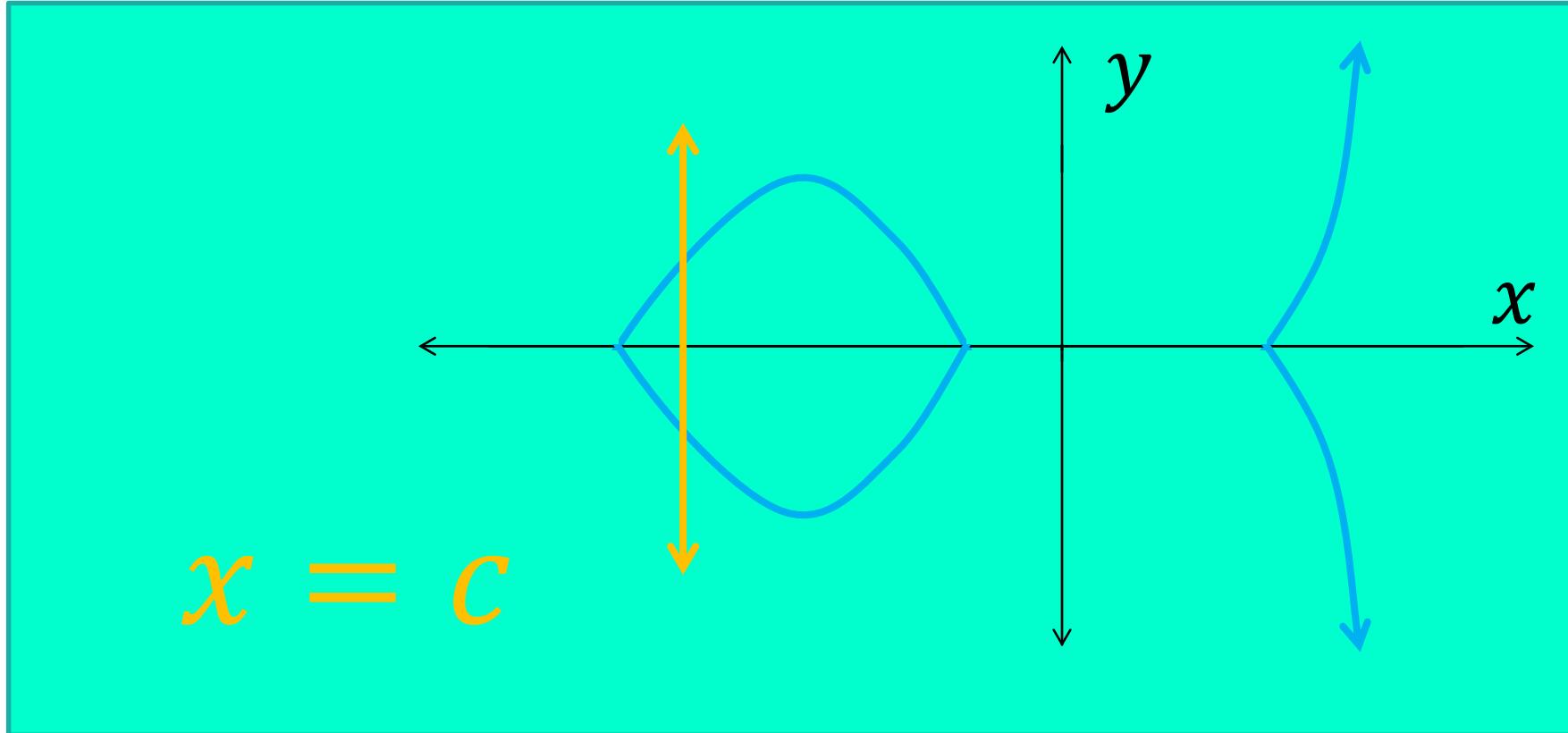
Elliptic Curves Intersecting Lines

Non-vertical Lines

- 1 intersection point (typical case)
- 2 intersection points (tangent case)
- 3 intersection points (typical case)

Elliptic Curves Intersecting Lines

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



Elliptic Curves Intersecting Lines

Vertical Lines

Elliptic Curve: $y^2 = x^3 + Ax + B$

Straight Line: $x = c$

$$y^2 = c^3 + Ac + B = C'$$

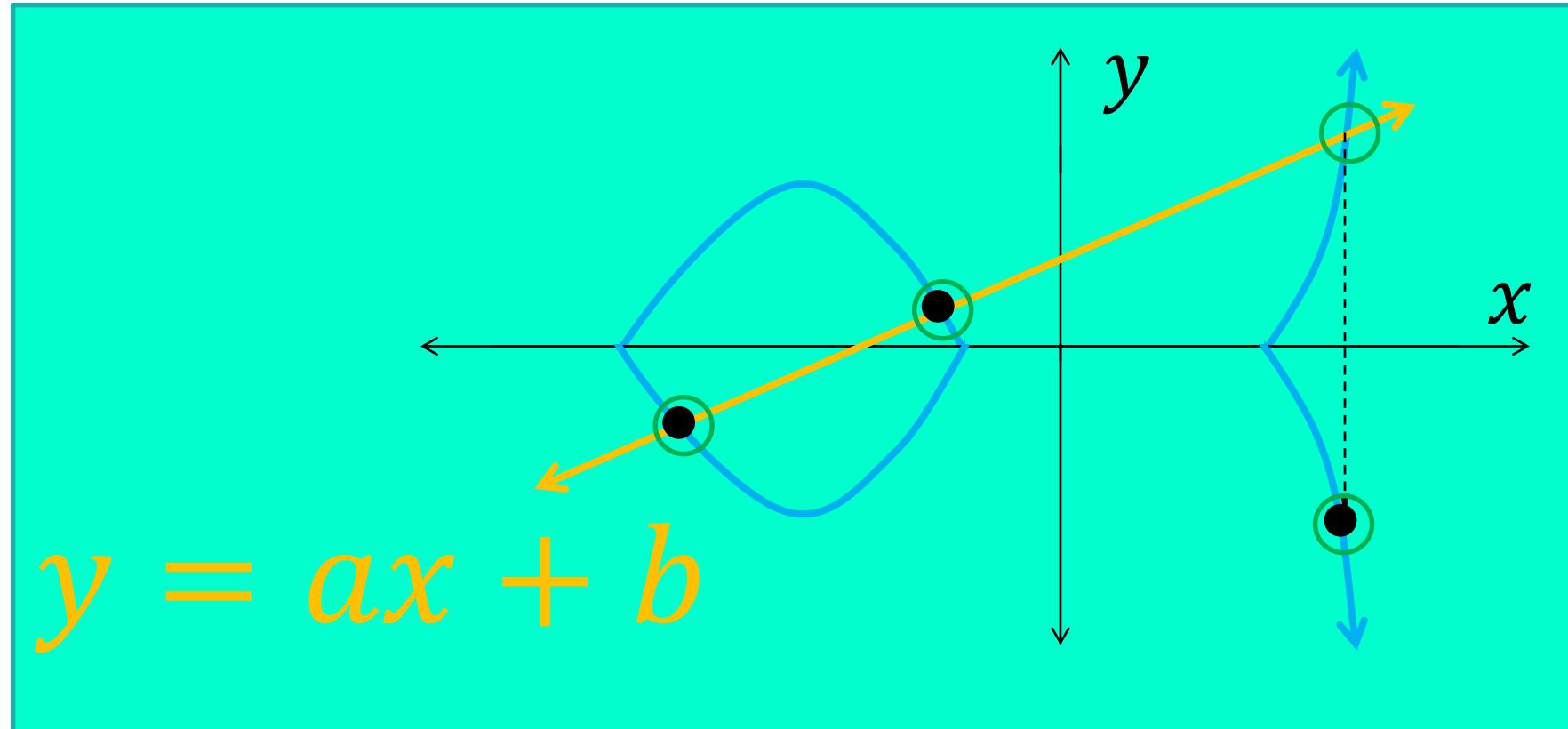
Elliptic Curves Intersecting Lines

Vertical Lines

- 0 intersection point (typical case)
- 1 intersection points (tangent case)
- 2 intersection points (typical case)

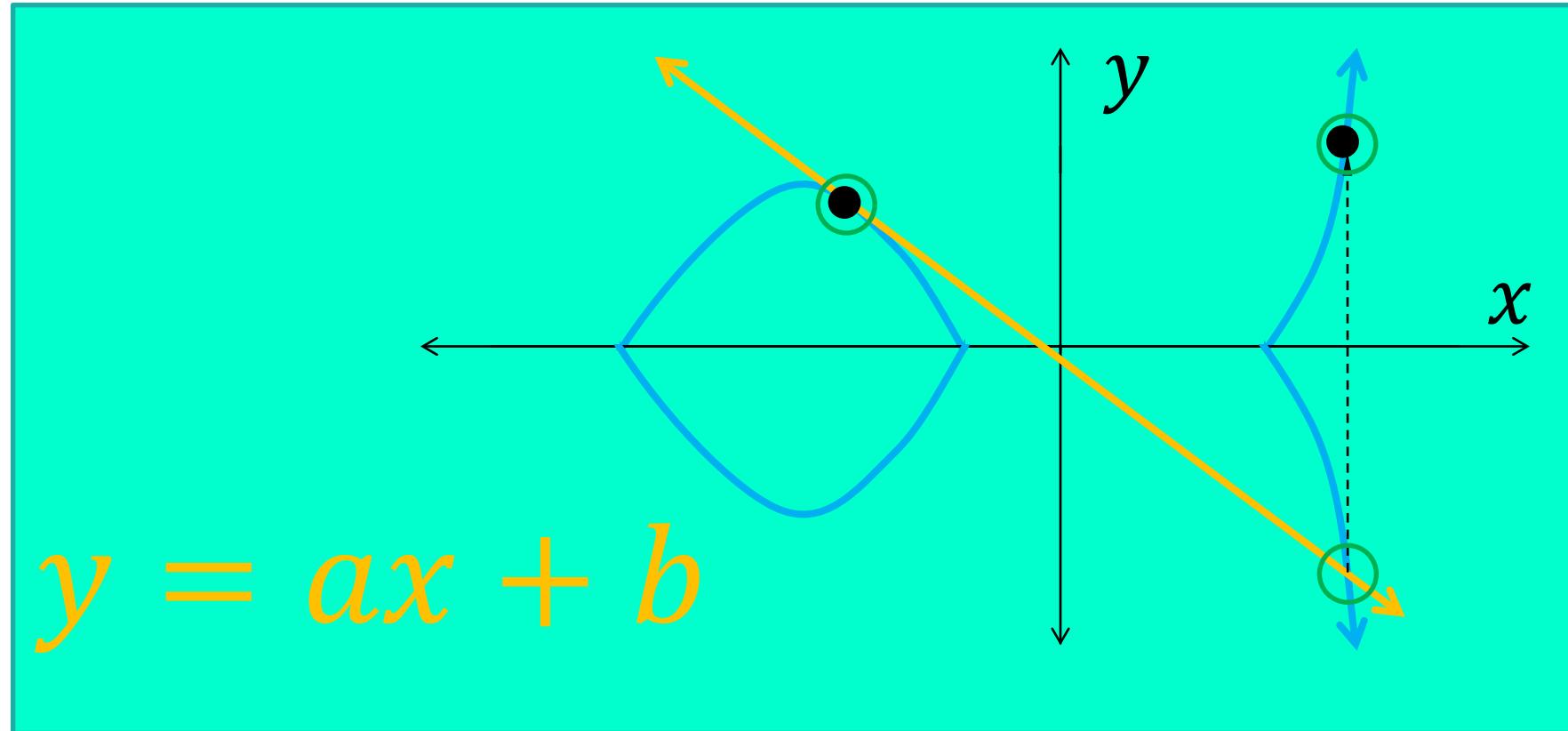
Elliptic Groups

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



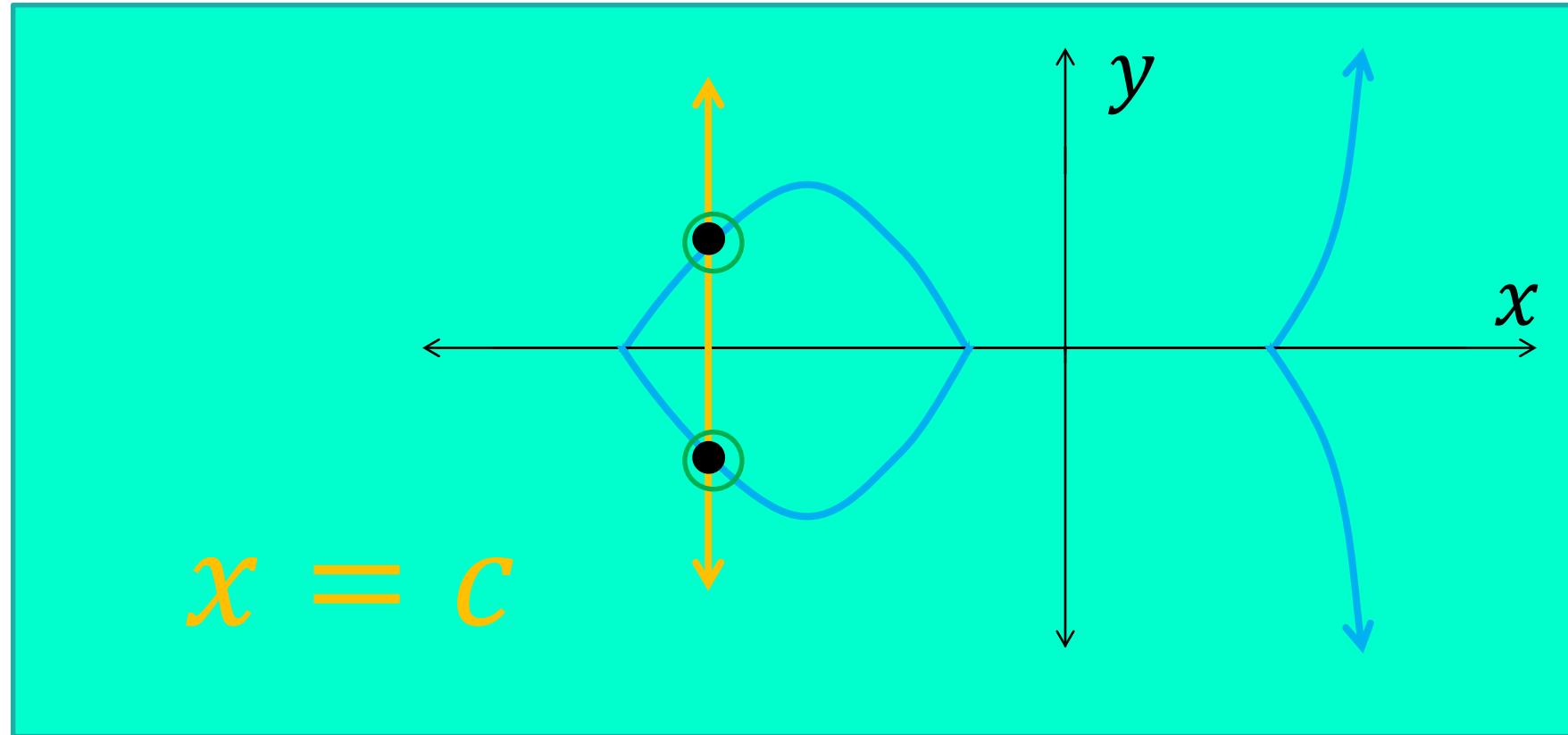
Elliptic Groups

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



Elliptic Groups

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

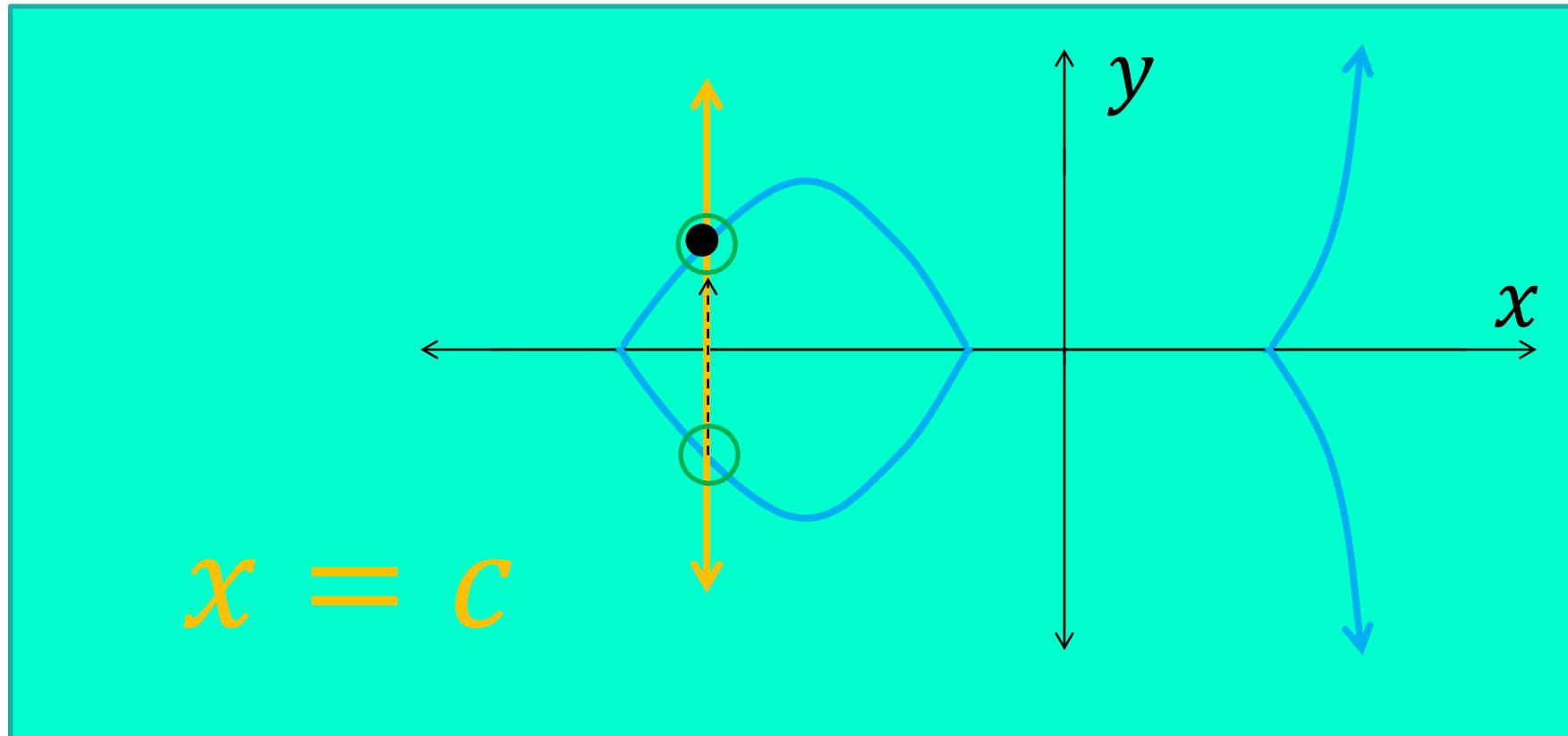


Elliptic Groups

- Add an “artificial” point **I** to handle the vertical line case.
- This point **I** also serves as the group identity value.

Elliptic Groups

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



Elliptic Groups

$$(x_1, y_1) \times (x_2, y_2) = (x_3, y_3)$$

when $x_1 \neq x_2$

$$x_3 = ((y_2 - y_1)/(x_2 - x_1))^2 - x_1 - x_2$$

$$y_3 = -y_1 + ((y_2 - y_1)/(x_2 - x_1))(x_1 - x_3)$$

Elliptic Groups

$$(x_1, y_1) \times (x_2, y_2) = (x_3, y_3)$$

when $x_1 = x_2$ and $y_1 = y_2 \neq 0$

$$x_3 = ((3x_1^2 + A)/(2y_1))^2 - 2x_1$$

$$y_3 = -y_1 + ((3x_1^2 + A)/(2y_1))(x_1 - x_3)$$

Elliptic Groups

$$(x_1, y_1) \times (x_2, y_2) = \mathbf{I}$$

when $x_1 = x_2$ but $y_1 \neq y_2$ or $y_1 = y_2 = 0$

$$(x_1, y_1) \times \mathbf{I} = (x_1, y_1) = \mathbf{I} \times (x_1, y_1)$$

$$\mathbf{I} \times \mathbf{I} = \mathbf{I}$$

Computation in Elliptic Groups

- For any two points u and v in elliptic group, we can now compute uv .
- For any point x in an elliptic group and any integer r , we can compute x^r .
- Large exponentiations can be computed efficiently by repeated squaring:

$$x^{1024} = (((((((((x^2)^2)^2)^2)^2)^2)^2)^2)^2$$

Example:

$$x^{360} = x^{256} \cdot x^{64} \cdot x^{32} \cdot x^8$$

Finite Elliptic Groups

Once we have everything written out algebraically, we can discard the geometry and work over a finite field by doing all computations modulo a prime $p > 3$.

$E_p(A, B)$ refers to the elliptic group defined by the elliptic curve $y^2 = x^3 + Ax + B$ with base arithmetic operations performed modulo p .

The Fundamental Equation

$$Z = Y^X \bmod N$$

The Fundamental Equation

$$Z = Y^X \text{ in } E_p(A, B)$$

The Fundamental Equation

$$Z = Y^X \text{ in } E_p(A, B)$$

When Z is unknown, it can be efficiently computed by repeated squaring.

The Fundamental Equation

$$Z = Y^X \text{ in } E_p(A, B)$$

When X is unknown, this version of the discrete logarithm is believed to be quite hard to solve.

The Fundamental Equation

$$Z = Y^X \text{ in } E_p(A, B)$$

When Y is unknown, it *can* be efficiently computed by “sophisticated” means.

Diffie-Hellman Key Exchange

Alice

- Randomly select a large integer a . Send $A = Y^a \text{ mod } N$.
- Compute the key $K = B^a \text{ mod } N$.

Bob

- Randomly select a large integer b . Send $B = Y^b \text{ mod } N$.
- Compute the key $K = A^b \text{ mod } N$.

$$B^a = Y^{ba} = Y^{ab} = A^b$$

Diffie-Hellman Key Exchange

Alice

- Randomly select a large integer a . Send $A = Y^a$ in $E_p(A, B)$.
- Compute the key $K = B^a$ in $E_p(A, B)$.

Bob

- Randomly select a large integer b . Send $B = Y^b$ in $E_p(A, B)$.
- Compute the key $K = A^b$ in $E_p(A, B)$.

$$B^a = Y^{ba} = Y^{ab} = A^b$$

Why use Elliptic Curves?

- The best *currently known* algorithm for EC discrete logarithms would take about as long to find a 160-bit EC discrete log as the best *currently known* algorithm for integer discrete logarithms would take to find a 1024-bit discrete log. Currently, 256-bit EC discrete logs take about as long as 3072-bit integer discrete logs.
- 256-bit EC algorithms are faster and use much shorter keys than 2048-bit “traditional” algorithms.

Why *not* use Elliptic Curves?

- EC discrete logarithms have been studied far less than integer discrete logarithms.
- There's no fundamental reason why a sub-exponential algorithm for EC discrete logarithms couldn't be found.
- Basic EC operations are more cumbersome than integer operations, so EC is only faster if the keys are *much* smaller.
- Getting the best performance from elliptic curves requires use of *special* curves.
- EC Cryptography requires the use of “sophisticated” pre-agreed curves.

Quantum Computing

What is a Quantum Computer?

A *quantum computer* operates on quantum mechanical *qubits* instead of ordinary binary bits.

An ordinary bit is either a 0 or a 1.

A qubit can hold the *superposition* of many ordinary bits.

What can a Quantum Computer Do?

Quantum computers can do two things better than ordinary computers.

1. Grover's Algorithm: Fast exhaustive search
 N objects can be searched in \sqrt{N} time.
2. Shor's Algorithm: Fast integer factorization
An N -bit integer can be factored in $poly(N)$ time.
(The same is true for discrete logarithms.)

Decoherence

- Keeping qubits stable is very, very hard.
- The standard trick is to use error correction – lots of error correction.
- But this requires more qubits.
- It can take tens of thousands of *physical* qubits to from a single *logical* qubit.

Post-Quantum Cryptography

What would cryptographers do in the face of a large quantum computer?

- Quantum cryptography is *not* the answer to quantum computation.
- The remedy is *quantum-resistant* (or *post-quantum*) cryptography.

Post-Quantum (PQ) Cryptography

Symmetric Ciphers

- Grover's algorithm enables searching through N objects in \sqrt{N} time.
- A classic attack on a good 128-bit cipher requires $\mathcal{O}(2^{128})$ time.
- A quantum attack can reduce this to $\mathcal{O}(2^{64})$ time.
- Solution: Use 256-bit ciphers.

Post-Quantum (PQ) Cryptography

Asymmetric Ciphers

- Shor's algorithm hits RSA, Diffie-Hellman and related asymmetric ciphers much harder.
- Quantum-resistant alternatives generally require *much* larger keys, *much* larger ciphertexts and signatures, or both.

One-Way Hash Functions

One more tool: One-Way Hash Functions

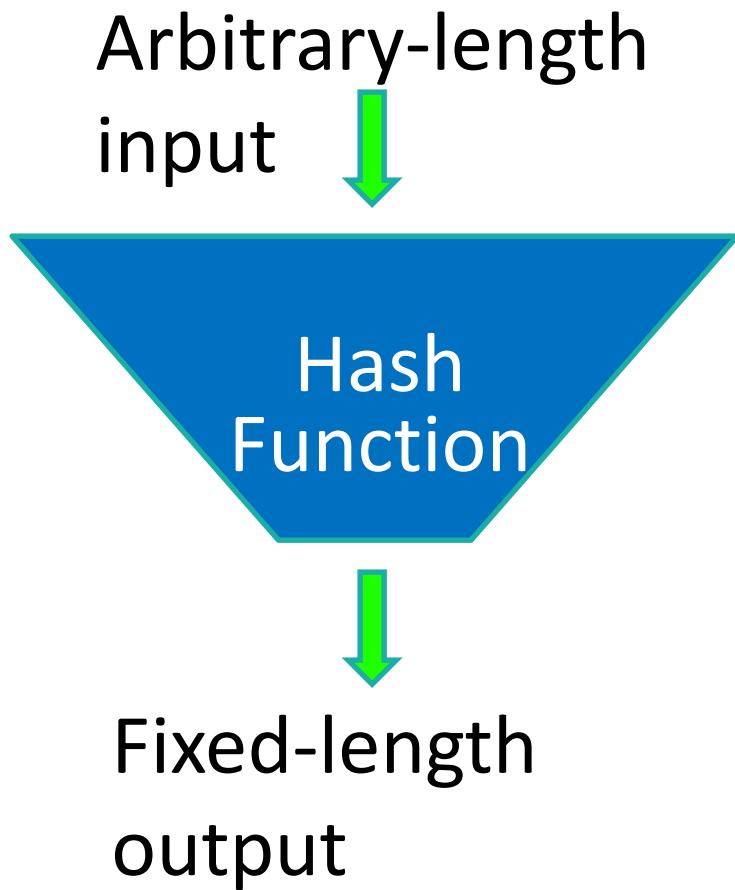
A *One-Way Hash Function* takes an arbitrary-length input and produces a “unique” fixed-length summary as its output.

Depending of the OWHF, the output is typically at least **256** bits, although some older OWHFs have shorter outputs.

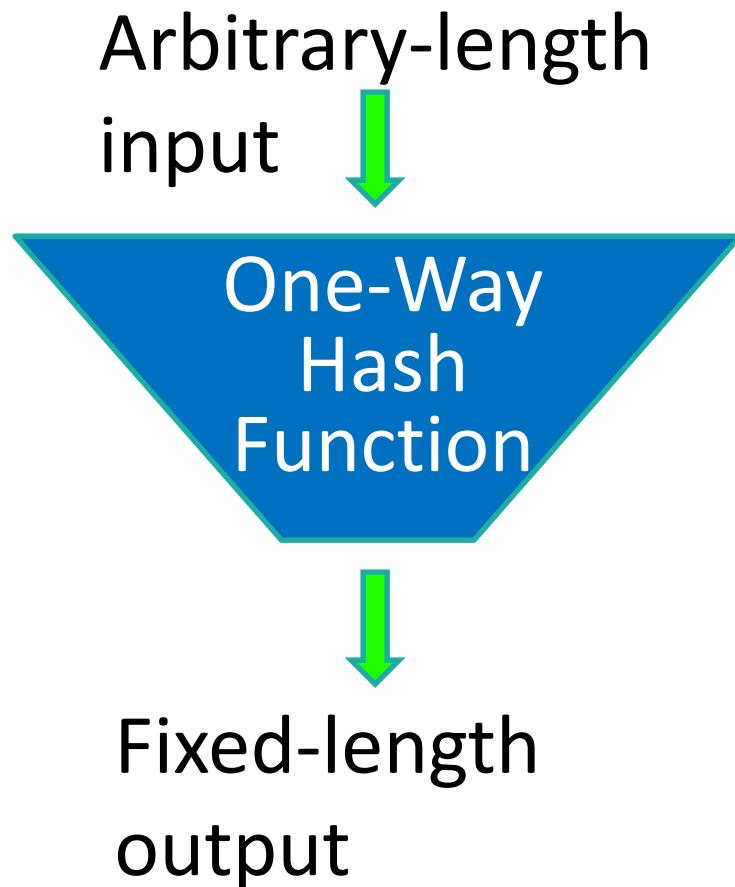
Some One-Way Hash Functions

- MD4, MD5 – 128-bit output
- SHA-0, SHA-1 – 160-bit output
- SHA-2 family: SHA-224, SHA-256, SHA-384, SHA-512
- SHA-3 family: variable

Hash Functions

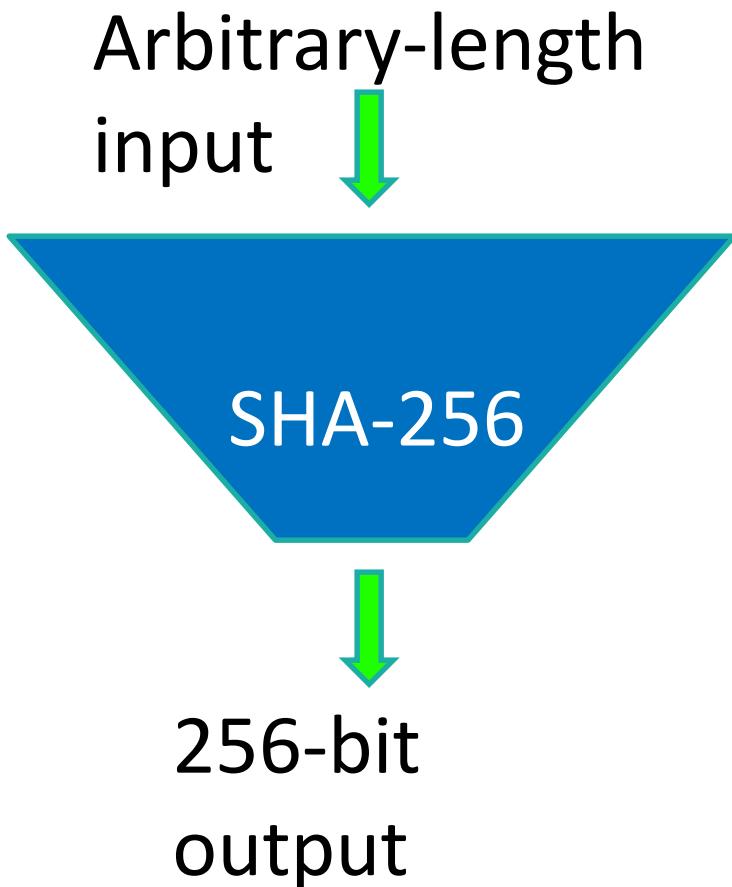


One-Way Hash Functions



Given just an output, it is infeasible to find *any* input which produces that output.

SHA-256



Given just an output, it is infeasible to find *any* input which produces that output.

3 Definitions of One-Way Hashes

1. Non-invertible: Given a target output, it's hard to find an input that produces the target output.
2. 2nd pre-image resistant: Given an input, it's hard to find another input that produces the same output.
3. Collision-intractable: It's hard to find *any* two inputs that produce the same output.

Birthday Paradox

If you pick more than $\sim\sqrt{N}$ times from a space of N items, you will start seeing repetitions.

Intuition: After K items have been seen,
$$\frac{K(K-1)}{2} \approx K^2$$
 pairs will have been seen.

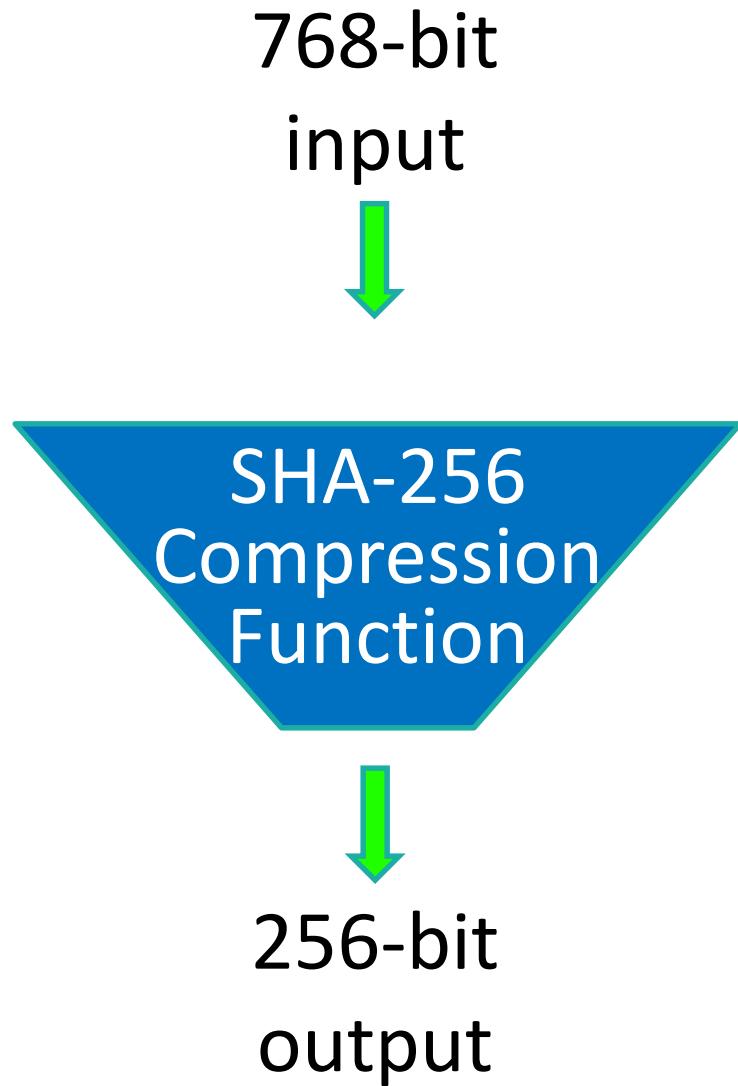
Some One-Way Hash Function Applications

A hash of data can serve as a *Message Digest* which can be signed in lieu of a full message.

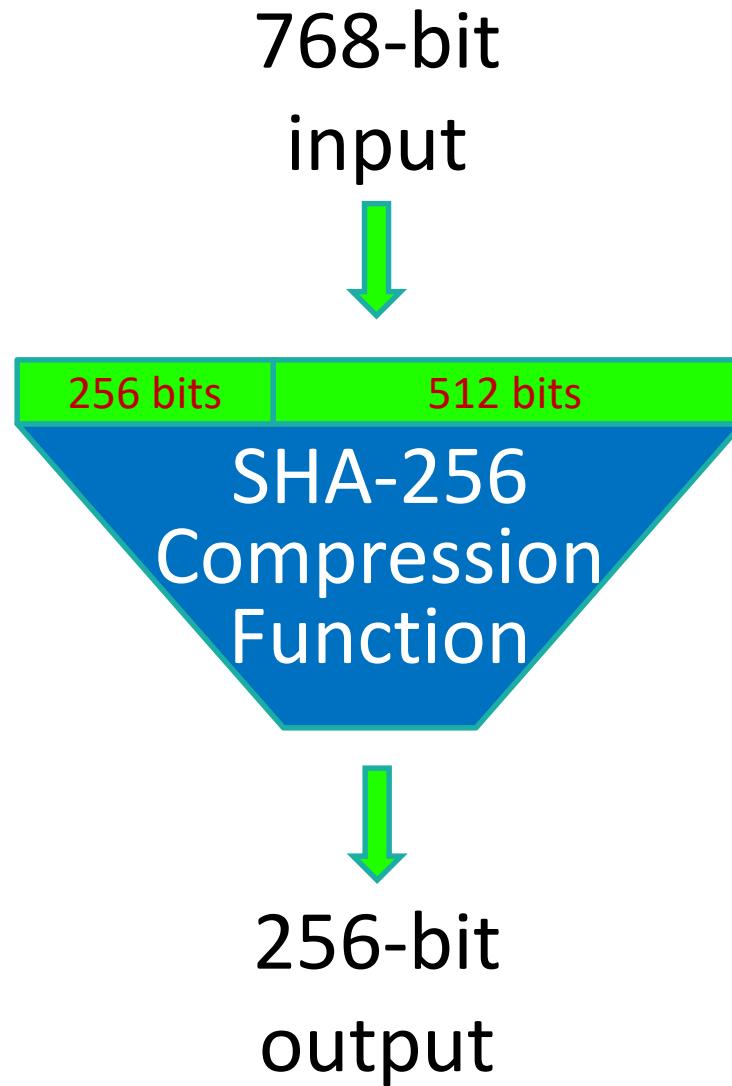
A keyed hash function can serve as a *Message Authentication Code (MAC)* to provide integrity.

SHA-256 Mechanics

SHA-256 Mechanics



SHA-256 Mechanics



SHA-256 Mechanics

Full Input

SHA-256 Mechanics

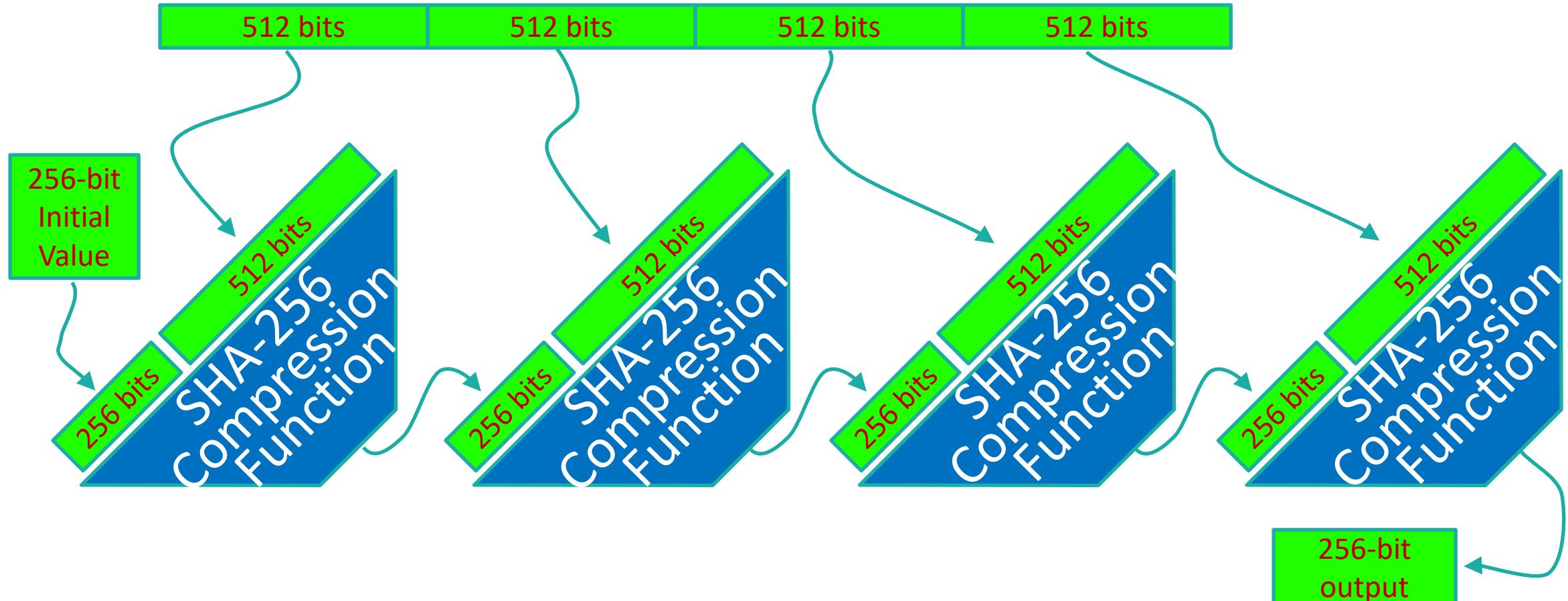
512 bits

512 bits

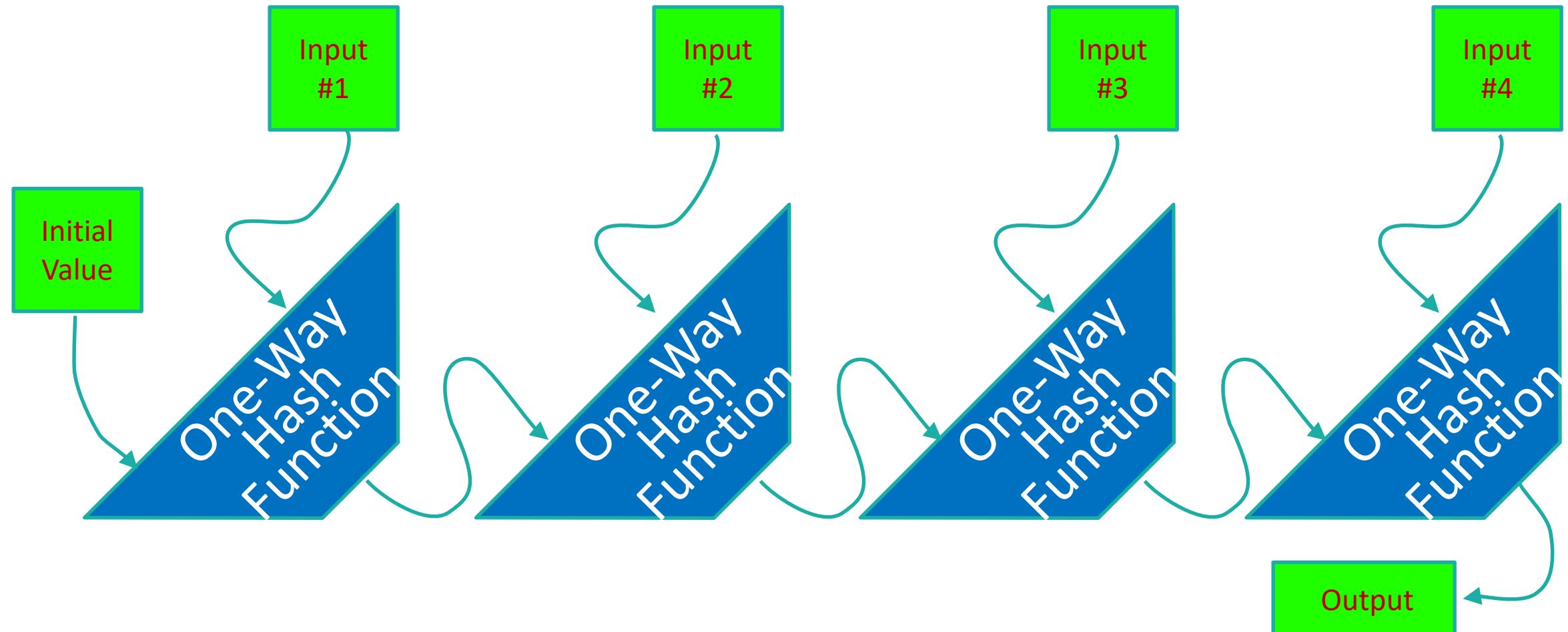
512 bits

512 bits

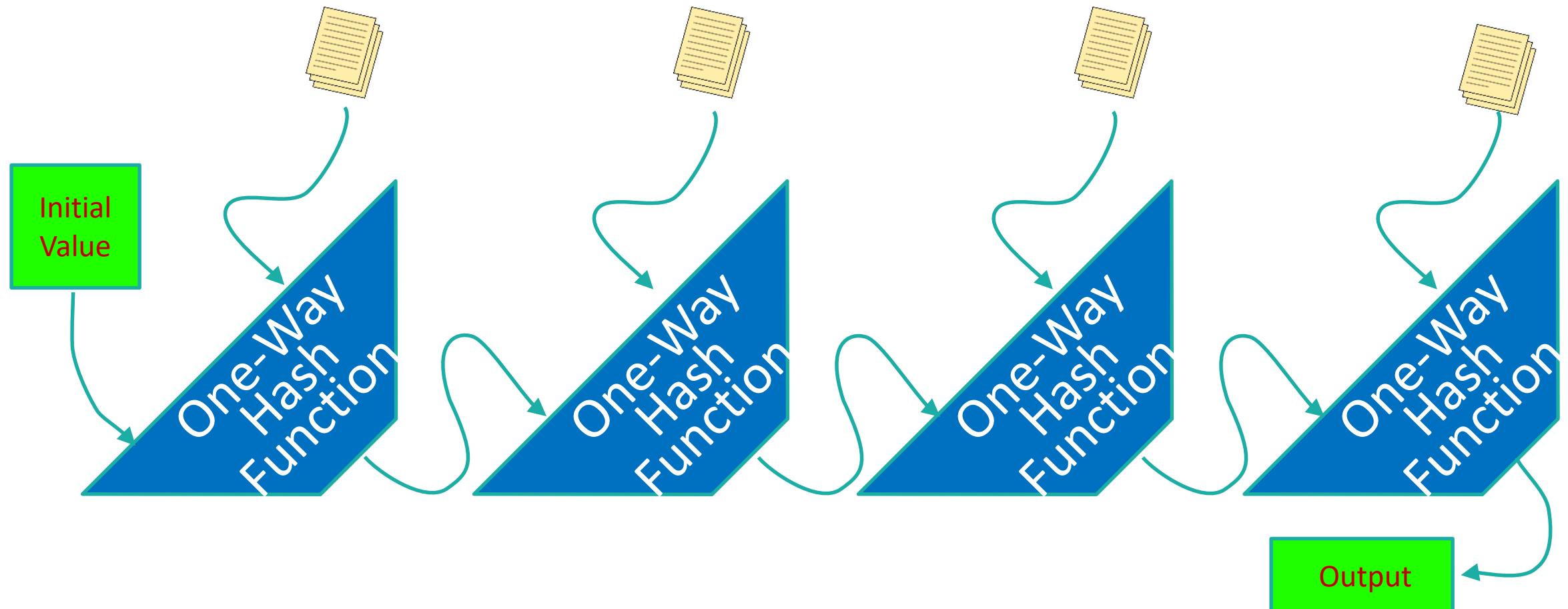
SHA-256 Mechanics



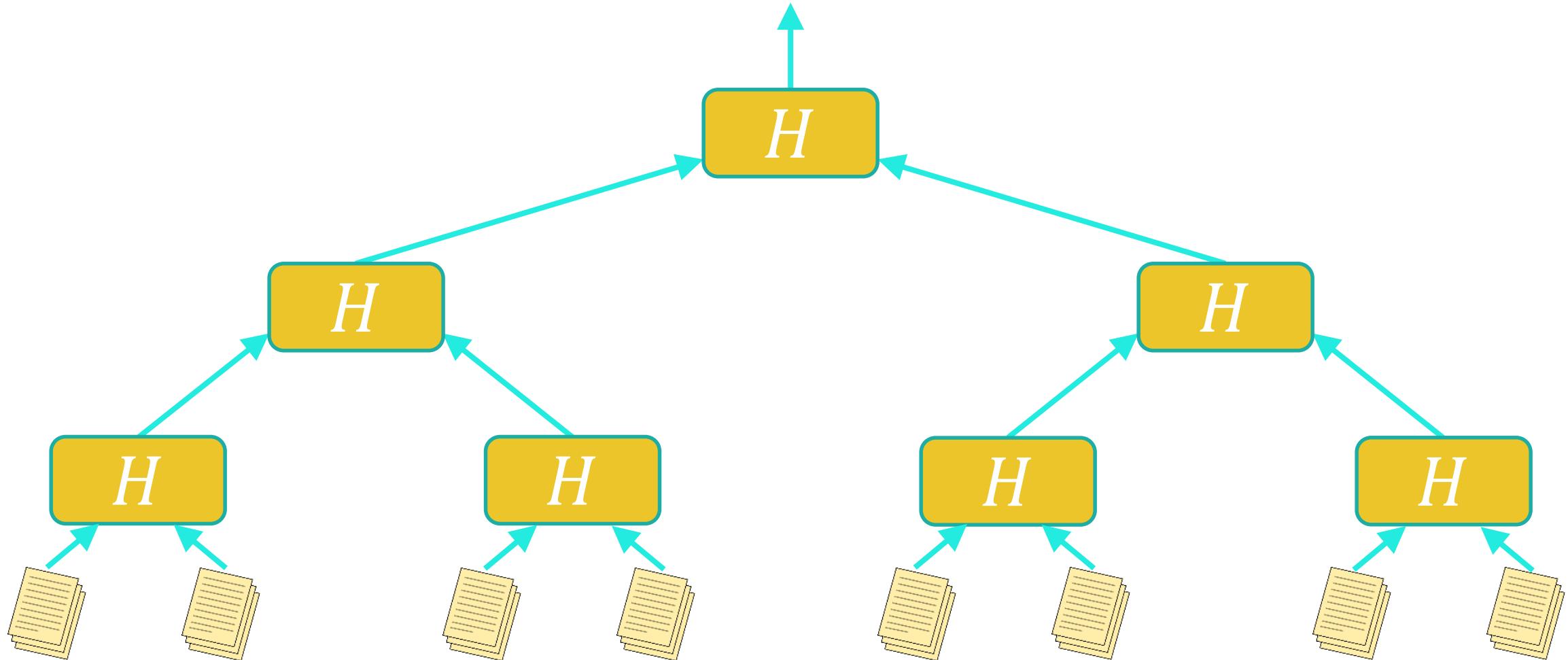
Hash Chains



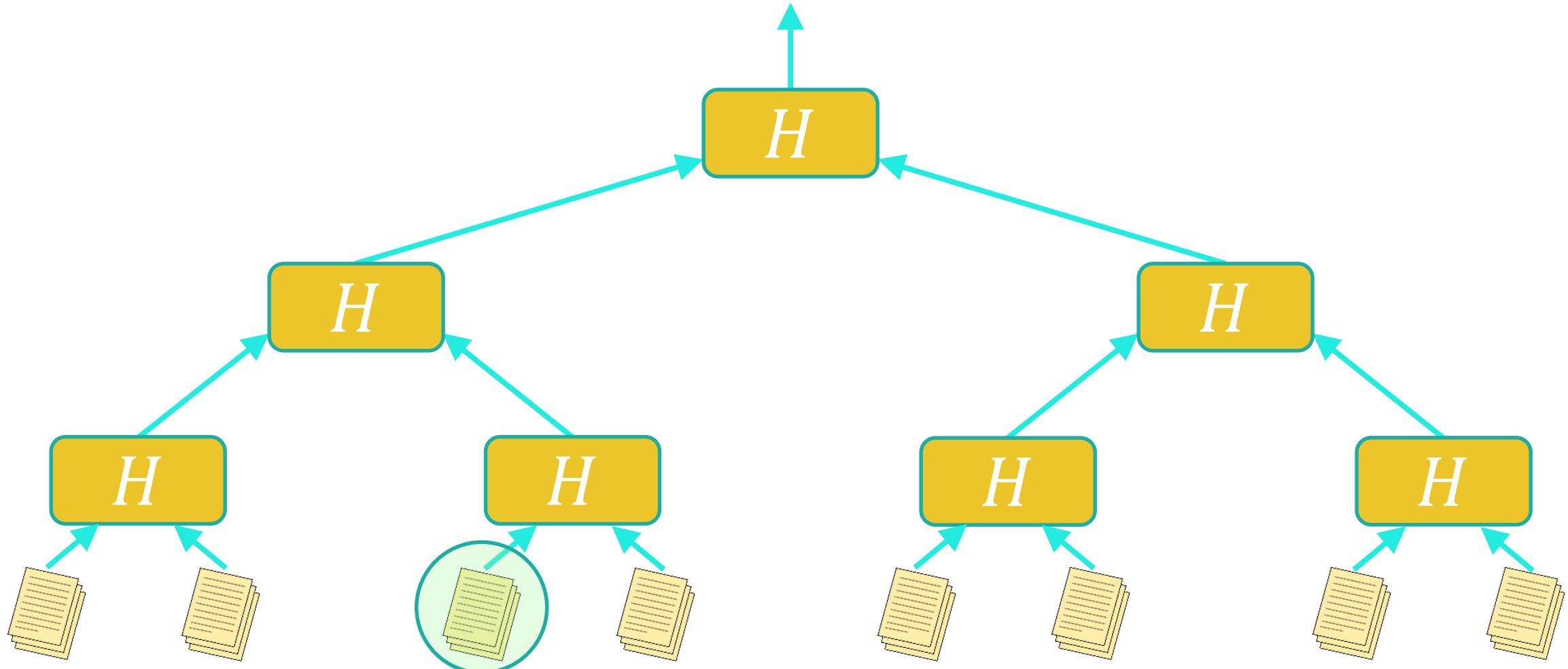
Hash Chains



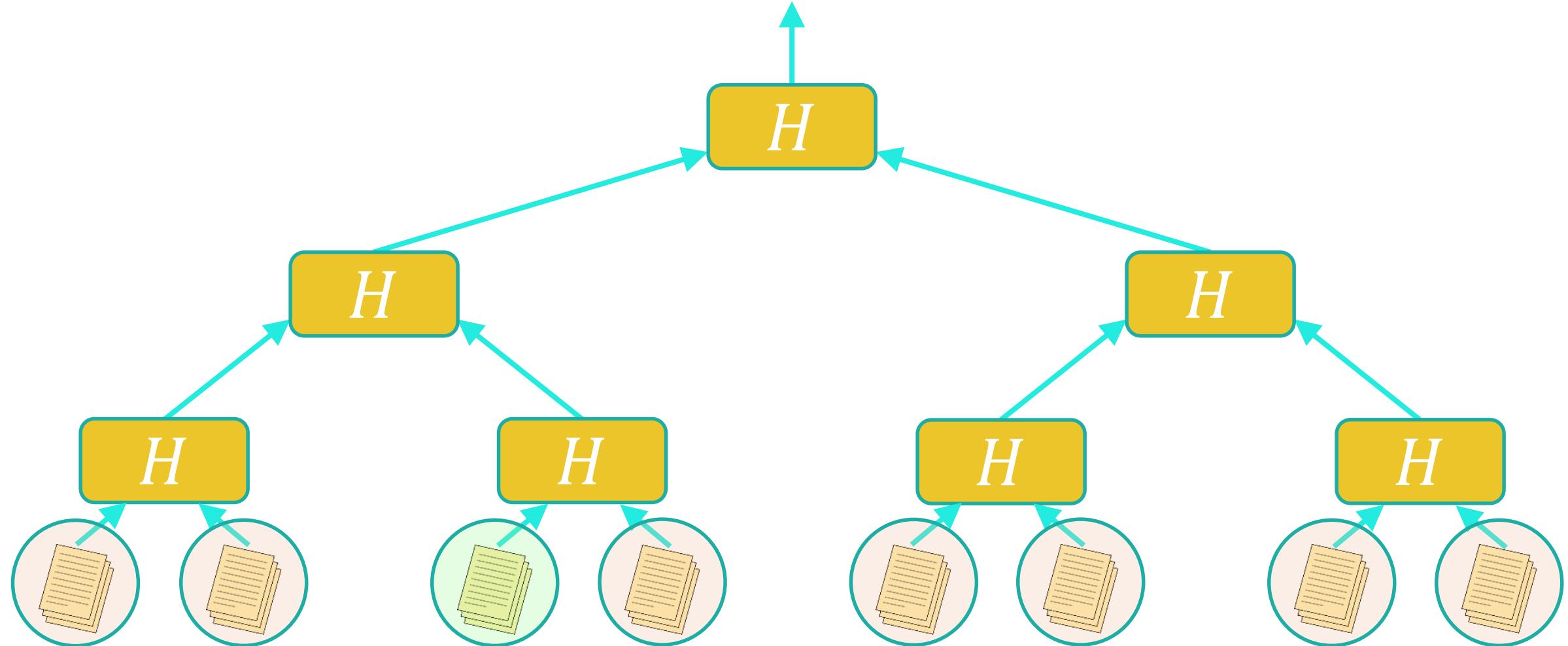
Merkle Tree



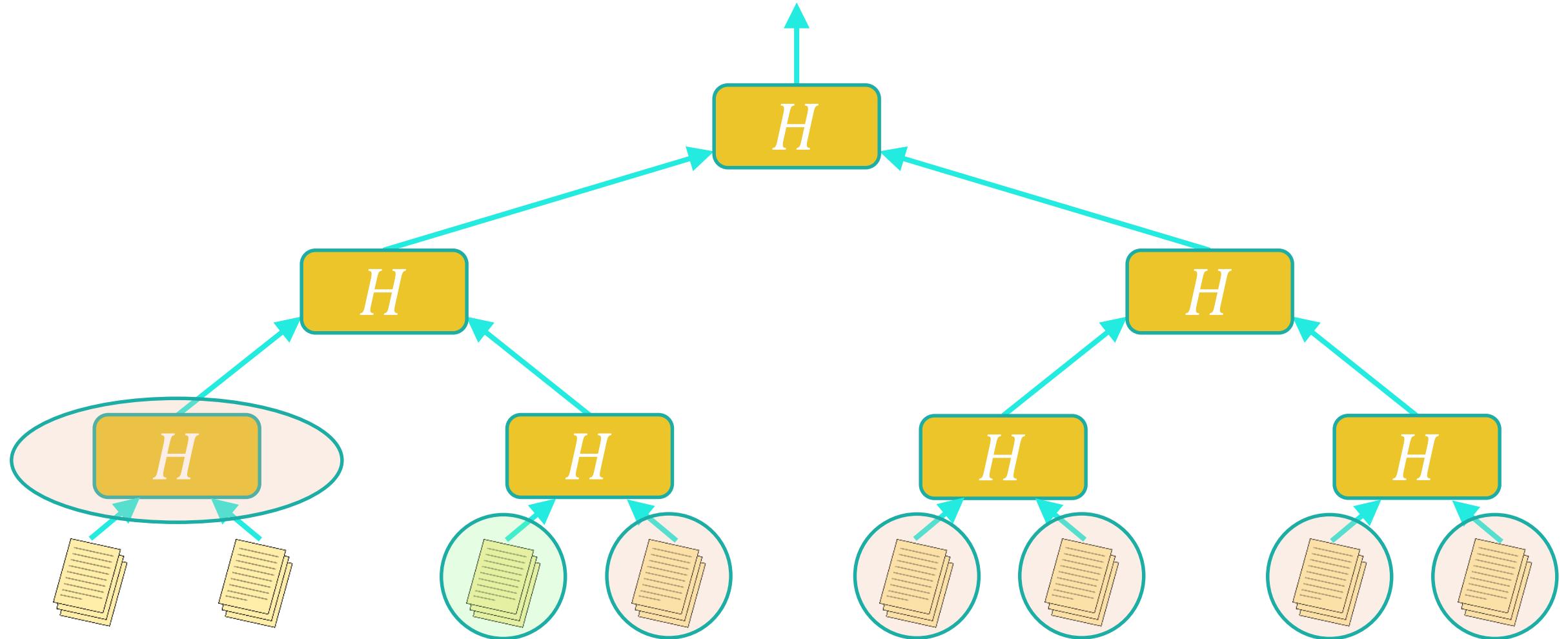
Merkle Tree



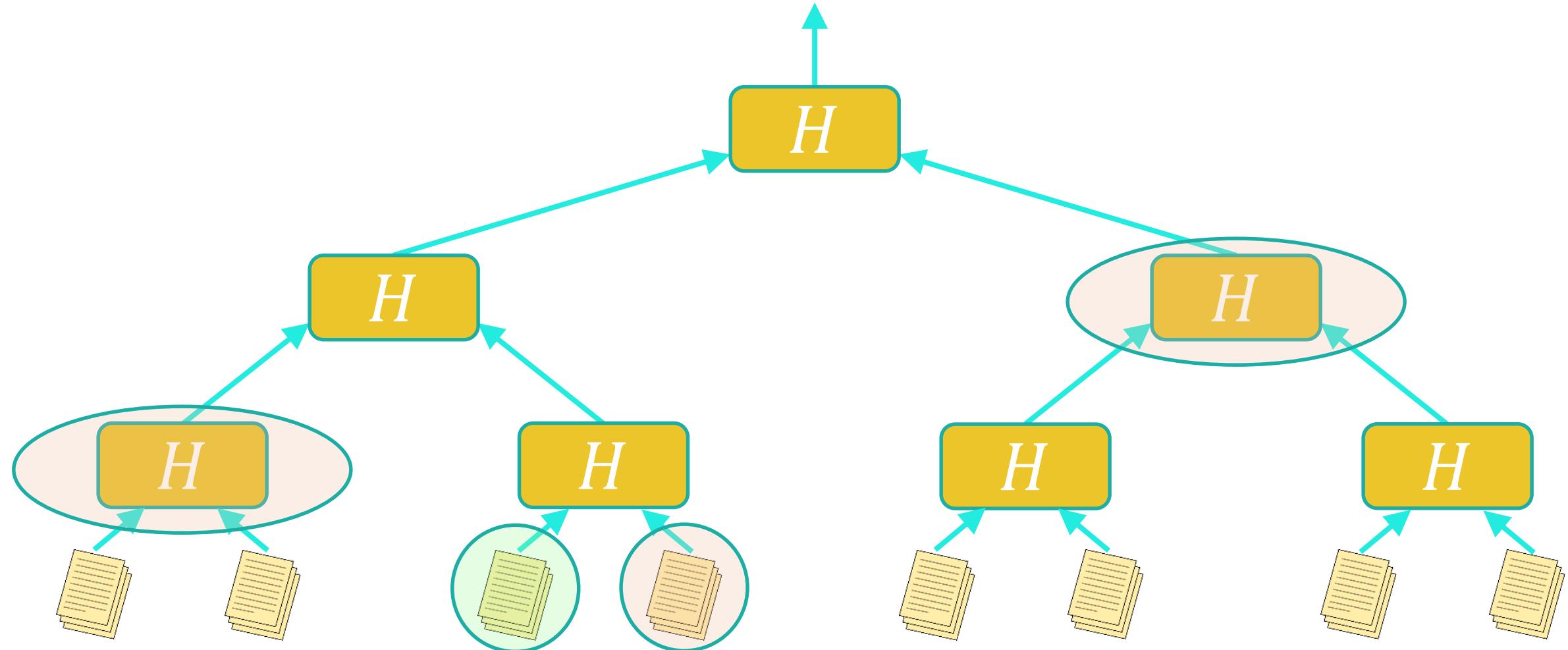
Merkle Tree



Merkle Tree



Merkle Tree



One-Way Accumulators

Flatten Merkle trees with constant-sized proof of membership.

Use a “*quasi-commutative*” function.

One-Way Accumulators

When viewed as a two-argument function, the (candidate) one-way function

$$F_N(Y, X) = Y^X \bmod N$$

also satisfies a useful additional property which has been termed *quasi-commutivity*:

$$F(F(Y, X_1), X_2) = F(F(Y, X_2), X_1)$$

since $Y^{X_1 X_2} = Y^{X_2 X_1}$.

One-Way Accumulators

The “hash” of values X_1, X_2, \dots, X_m is

$$Z = F_N(Y, X_1, X_2, \dots, X_m) = Y^{\sum_{i=1}^m X_i} \bmod N.$$

The value X_k can be shown to be one of the hashed values by showing $Z_k = Y^{\sum_{i \neq k} X_i} \bmod N$ since $Z = Z_k^{X_k} \bmod n$.

Historical Use of Hash Chains

- 1979: Merkle-Damgård – Chained Hash Function Construction
- 1979: Merkle Tree – Tree of Hashes used for Membership
- 1981: Lamport – Hash Chains for Password Protection
- 1991: Haber-Stornetta – Time-Stamping with Hash Trees
- 1994: Benaloh-deMare – One-Way Accumulators
- 1997: Hashcash – Generating “cash” by Repeated Hashing
- 2001: Rivest-Shamir – PayWord & MicroMint micropayments

Finding Distinguished Outputs

With SHA-256 (or any good one-way hash function) ...

- The best way to achieve a specific target output is to repeatedly try different inputs until one succeeds.
- The best way to achieve a specific output property is to repeatedly try different inputs until one succeeds.

Hashcash (1987)

To demonstrate work on x , find y such that

$$H(x, y) < z,$$

for some pre-determined bound z .

Hashcash – Proof of Work

To demonstrate work on x , find y such that

$$H(x, y) < z,$$

for some pre-determined bound z .

bitcoin Currency (2008)

The next *bitcoin(s)* are awarded to the first person who can find a value which when hashed with the previous *bitcoin* produces an output smaller than a pre-defined target.

bitcoin Transactions

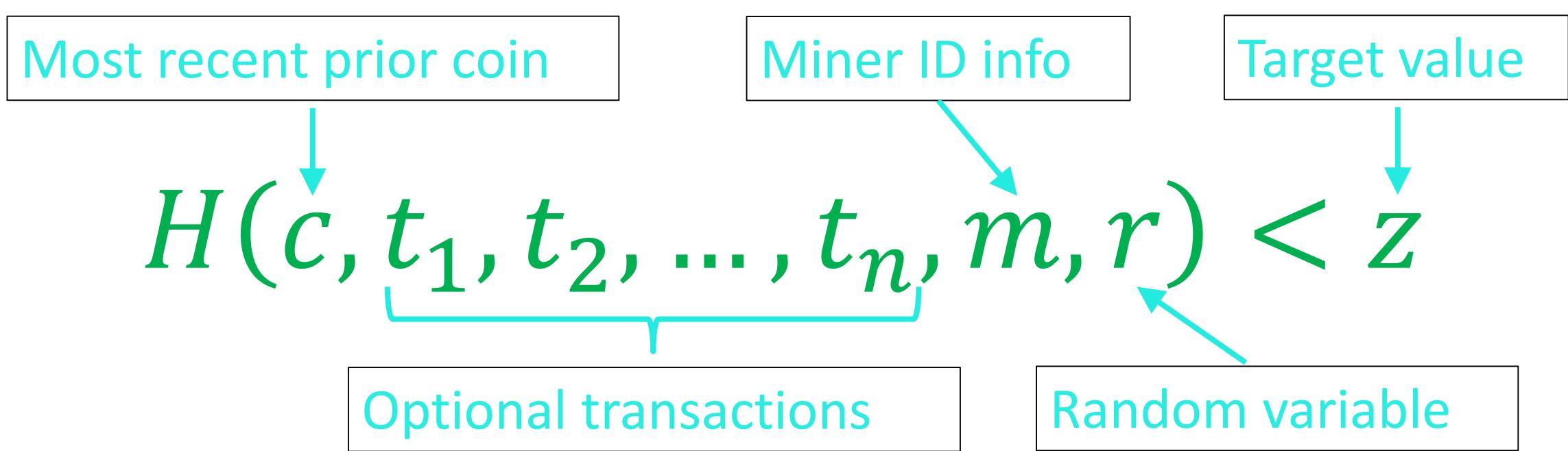
Together with the most recently found coin, a bitcoin *miner* can optionally include some signed “transactions” in its hashes (for which it may receive transaction fees).

bitcoin Transactions

The *values* that are hashed together with the previous coins can contain other *stuff*:

- My name
- My public key
- Transactions
- Contracts
- Etc.

bitcoin mining



bitcoin mining

- With bitcoin, the number of leading zeros required to find a “coin” adjusts automatically.
- Currently, that number is ~73.
- That means ~ $10,000,000,000,000,000,000$ one-way hashes must be performed to find a bitcoin.

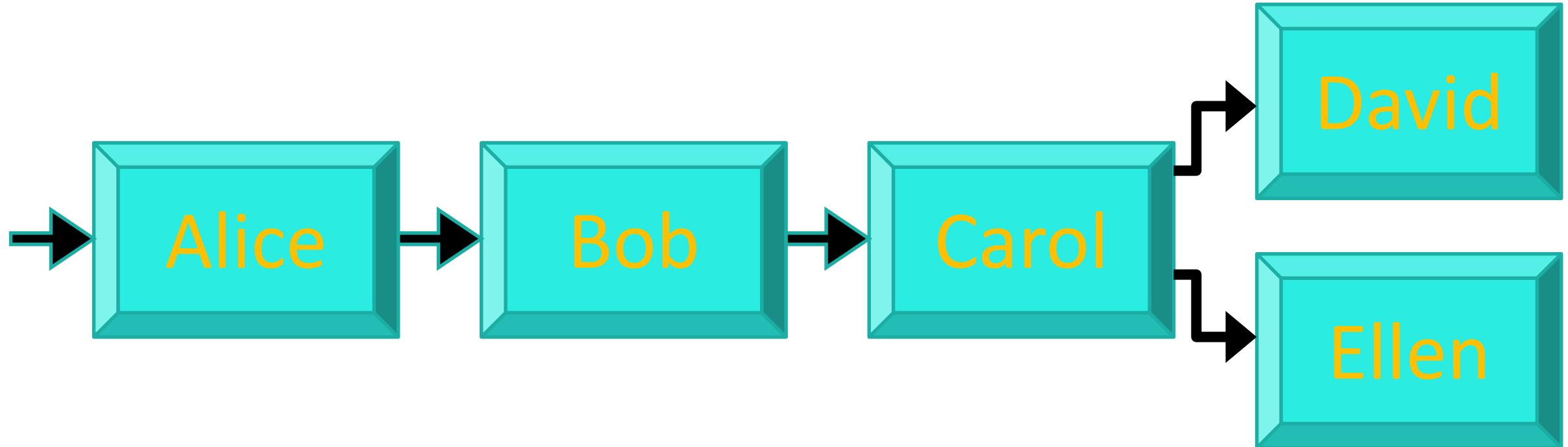
Mining new *bitcoins*

When a miner successfully extends the blockchain, it broadcasts its new value to all the other miners and receives a reward.

Miners then (are supposed to) continue mining on the new, longer chain.

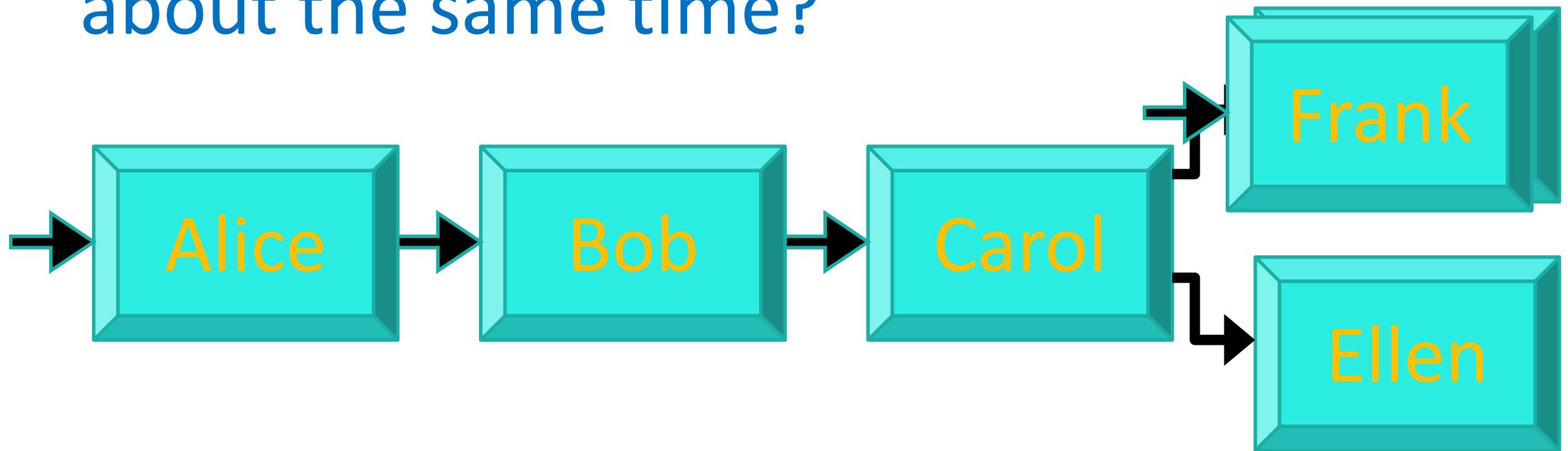
Resolving Conflicts

- What if two miners each find a block at about the same time?

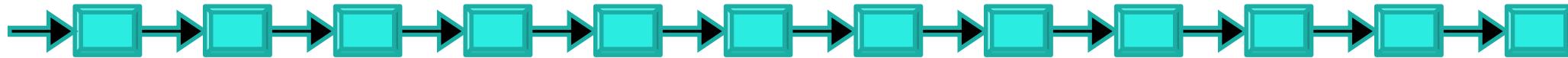


Resolving Conflicts

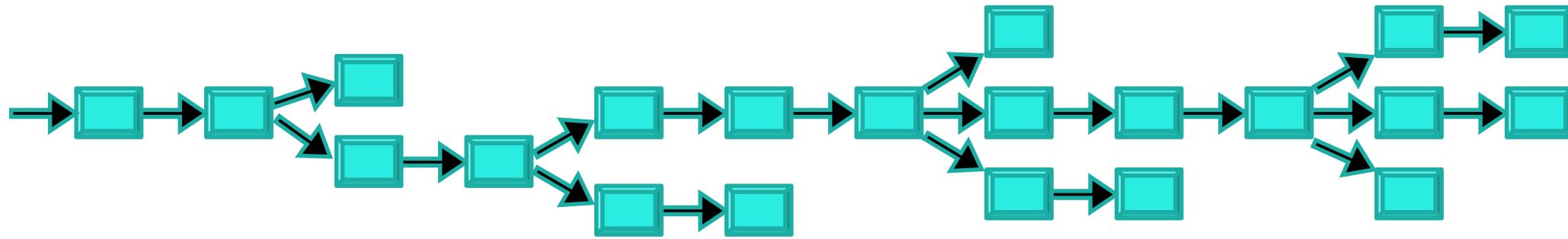
- What if two miners each find a block at about the same time?



“The” Blockchain



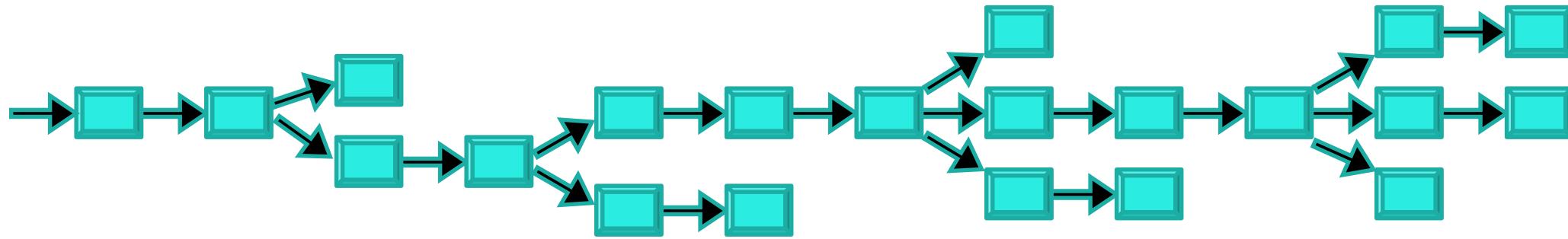
“The” Blockchain



“The” Blockchain

- You never *really* know whether your block/transaction is in “the” blockchain.
- Bitcoin convention is to wait until 6 blocks have been added to your block.
- A majority (or even large minority) can exclude blocks or hijack the blockchain.

“The” Blockchain



Mining Pools

Collaboration offers two principal benefits.

- Centralized transaction processing
- Reduced volatility

Pooling Details

- Pools can pay members in proportion to their failed contributions.
- Large pools threaten integrity of the system.

What Do Block Chains Provide?

- Block chains can achieve distributed consensus without a trusted authority or random source.
- Block chains can *randomly* select a leader/winner from a group in a “fair” manner.

Block Chain Overreach

- Block chains are *not* ideal when a central authority is already part of the system.
- N.B. Hash chains (now sometimes called *private* block chains) have numerous good applications.

Verifiable Elections

Blockchains and Elections

- Elections have central authorities.
 - Set the ballot contents
 - Set and maintain eligibility requirements
 - Set start/end time of election
- Note that authority need not be trusted!

Blockchains and Elections

An election's designated central authority can simply post the same information (digitally signed) on a public web site.

Blockchains and Elections

- Blockchains do not provide anonymity and authentication.
- These can be provided with cryptography.
- But once the cryptography is added, the blockchains become superfluous.

Blockchains and Elections

Blockchains don't solve fundamental problems with online voting.

- Client malware can change votes.
- Targeted DoS can disenfranchise voters.
- Voters are subject to coercion.

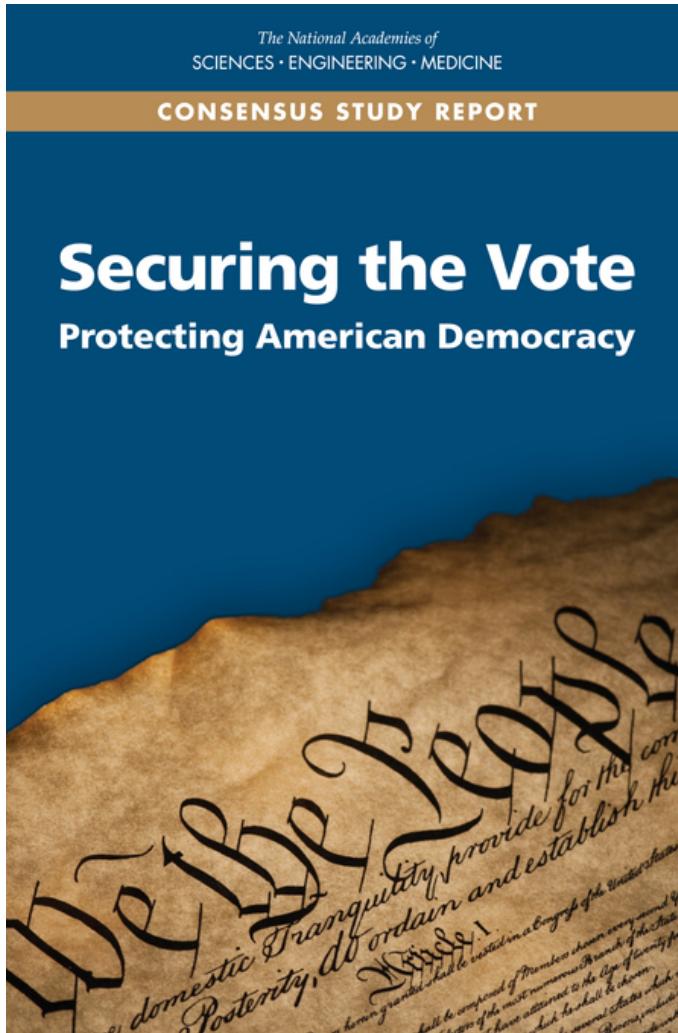
Blockchains and Elections

Blockchains create new problems.

- There is no accountability.
- There is no certainty.
- A mining majority has total control.

National Academies of Science, Engineering, and Medicine Study

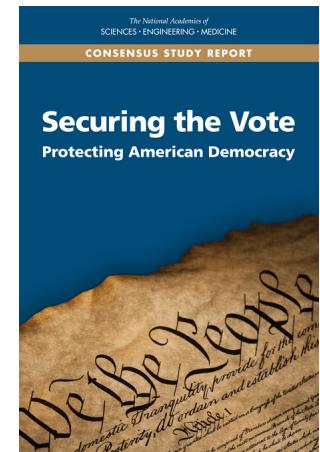
2016-2018



Findings and Recommendations

The election equipment market and certification process are badly broken.

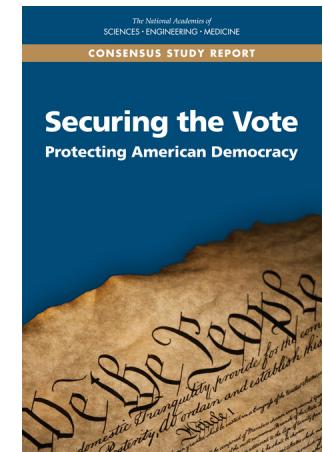
We need better ways to incentivize innovation.



Findings and Recommendations

Congressional funding should be provided to better support

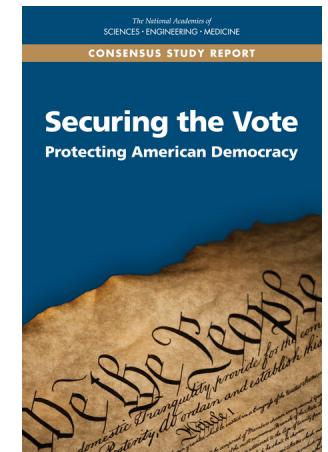
- Election Assistance Commission,
- state and local jurisdictions,
- research, and
- NIST standards (VVSG).



Findings and Recommendations

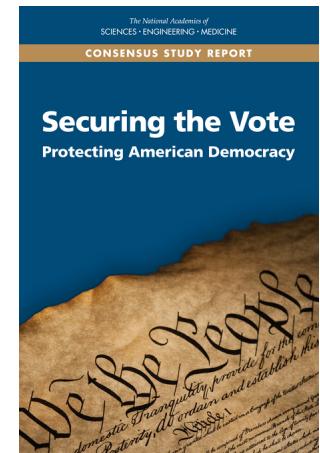
States should

- provide more election funding,
- participate in cross-state registration list matching programs, and
- provide vote-by-mail tracking.



Findings and Recommendations

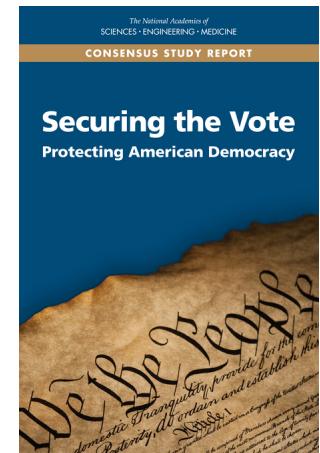
Internet Voting should not be done today
(and never with blockchains).



Findings and Recommendations

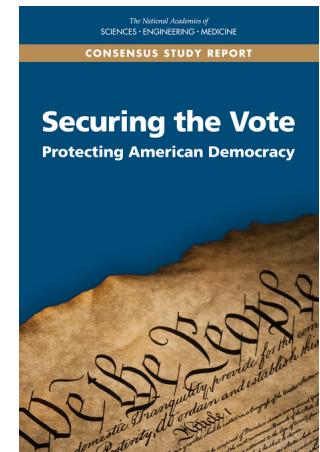
There was extensive Russian intrusion into the 2016 election in the form of disinformation and infiltration of voter registration databases.

However, there is no evidence of tampering with actual votes.



Findings and Recommendations

Nevertheless, the vote casting and tabulation systems are *extremely* vulnerable.

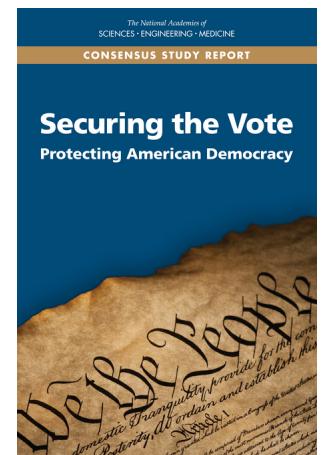


“My undergraduate security class could have changed the results of the 2016 election.”

Prof. J Alex Halderman
– University of Michigan

Findings and Recommendations

We should replace existing paperless voting systems with paper-based systems.

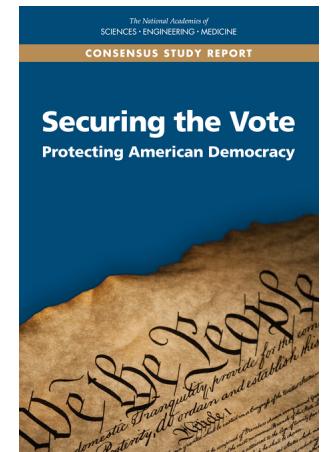


Findings and Recommendations

We must apply *best practices* to our election registration and voting systems.

However, this is not sufficient.

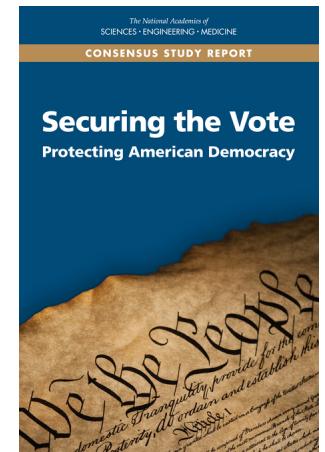
The challenge is asymmetric.



Findings and Recommendations

Since we can't ensure that our election systems cannot be corrupted, good auditing is essential.

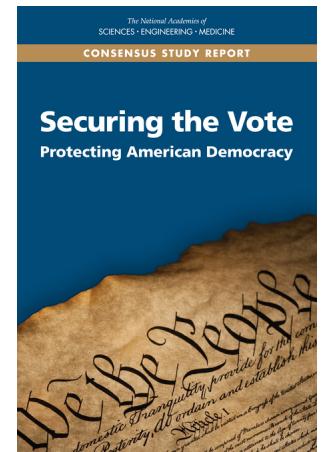
*We can at least detect tampering
– even if we can't prevent it.*



Findings and Recommendations

Two kinds of auditing are recommended.

1. Administrative: Risk-Limiting Audits
2. Public: End-to-End Verifiability



Findings and Recommendations

Advanced statistical methods and new techniques can enable far more efficient traditional administrative audits.

- Ballot-Polling Audits
- Ballot-Comparison Audits

Findings and Recommendations

Cryptographic techniques can enable public audits that shift the paradigm and democratize the electoral process.

What is Possible?

Technology exists that enables *any* inaccuracies and tampering of election tallies to be detected ...

... not just by **election officials**, but also by any **candidate, media outlet, voter, or other observer** ...

... and not just external tampering, but **corruption by election officials, equipment vendors, and others.**

This is known as *End-to-End (E2E) Verifiability*.

Findings and Recommendations

An election is *end-to-end verifiable* if

1. Voters can verify that their own selections have been correctly recorded.
2. Anyone can verify that the recorded votes have been correctly tallied.

End-to-End Verifiability

So called, *End-to-End (E2E) Verifiability* is the answer to the question

How can I trust the accuracy of an election outcome ...

when I don't trust the software, hardware, or personnel responsible for conducting the election?

An E2E-Verifiable Election

Voter Name	Vote
Alice Smith	Jefferson
Bob Williams	Adams
Carol James	Adams
David Fuentes	Jefferson
Ellen Chu	Jefferson

Totals	
Jefferson	3
Adams	2

End-to-End Verifiability

Note that end-to-end verifiability is a property of an individual election – not election equipment or systems.

An E2E-Verifiable Election

Voter Name	Vote
Alice Smith	Jefferson
Bob Williams	Adams
Carol James	Adams
David Fuentes	Jefferson
Ellen Chu	Jefferson

Totals	
Jefferson	3
Adams	2

Secret-Ballot

Most U.S. presidents were elected *without* the benefit of secret ballots.



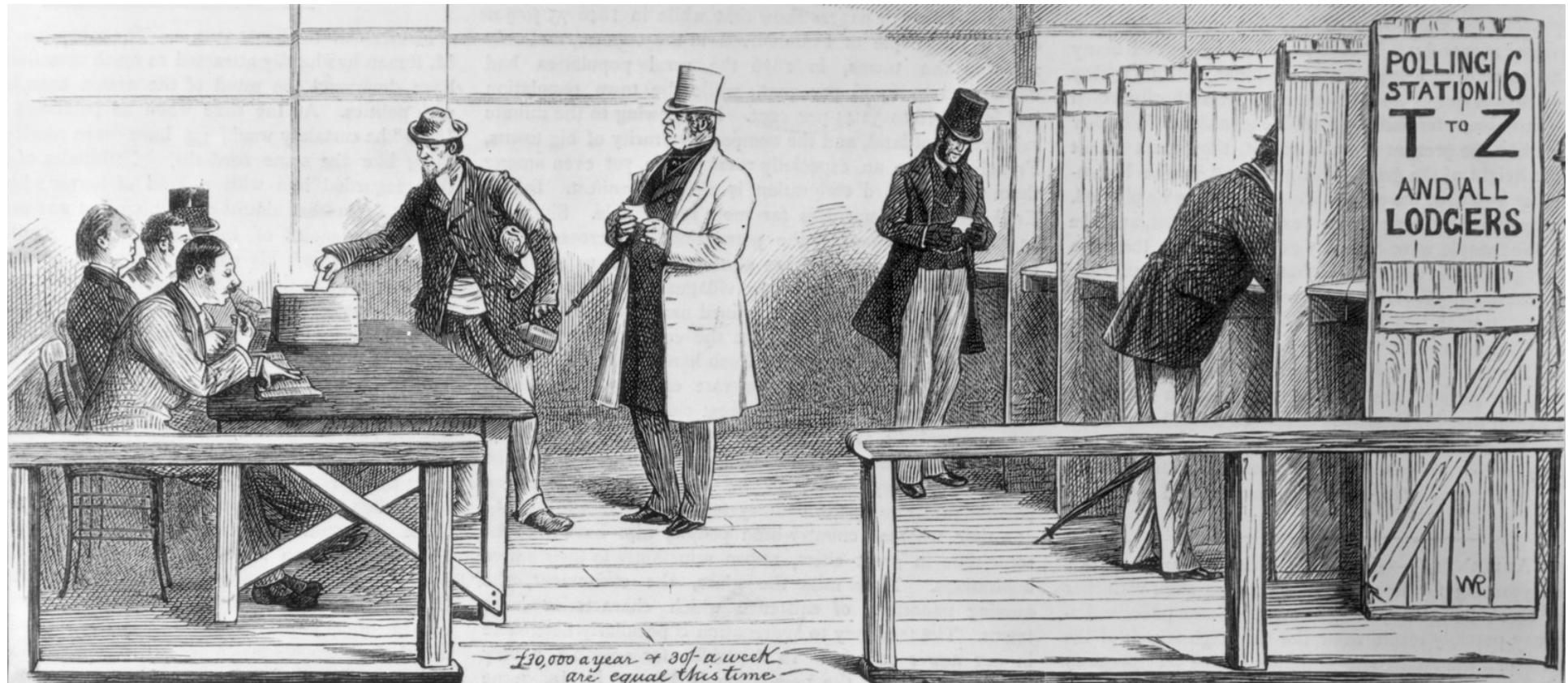
The County Election – George Caleb Bingham (1854)

Secret-Ballot

Most U.S. presidents were elected *without* the benefit of secret ballots.

Prior to the mid-19th century, we did not have the technical acumen to protect voter privacy.

The Australian Ballot



Secret-Ballot

Most U.S. presidents were elected *without* the benefit of secret ballots.

Prior to the mid-19th century, we did not have the technical acumen to protect voter privacy.

It was the 1888 election of Benjamin Harrison over incumbent Grover Cleveland in which voter coercion and vote-buying were so substantial that by 1892 led to the widespread adoption of secret ballots in the U.S.

Privacy

The only ingredient missing from this *transparent* election is privacy – and the things which flow from privacy (e.g. protection from coercion).

Performing tasks while preserving privacy is the bailiwick of cryptography.

Cryptographic techniques can enable E2E-verifiable elections while preserving voter privacy.

Adding Encryption

A layer of confidentiality can be added to an otherwise openly-verifiable system by encrypting the actual votes.

An E2E-Verifiable Election

Voter Name	Vote
Alice Smith	Jefferson
Bob Williams	Adams
Carol James	Adams
David Fuentes	Jefferson
Ellen Chu	Jefferson

Totals	
Jefferson	3
Adams	2

An E2E-Verifiable Election

Voter Name	Vote	
Alice Smith	Jefferson	X37BM6YPM
Bob Williams	Adams	2J8CNF2KQ
Carol James	Adams	VRSF5JQWZ
David Fuentes	Jefferson	MW5B2VA7Y
Ellen Chu	Jefferson	8VPPS2L39

Totals	
Jefferson	3
Adams	2

An E2E-Verifiable Election

Voter Name	Vote	
Alice Smith	Jefferson	X37BM6YPM
Bob Williams	Adams	2J8CNF2KQ
Carol James	Adams	VRSF5JQWZ
David Fuentes	Jefferson	MW5B2VA7Y
Ellen Chu	Jefferson	8VPPS2L39

Totals	
Jefferson	3
Adams	2

An E2E-Verifiable Election

Voter Name	Vote	
Alice Smith	Jefferson	X37BM6YPM
Bob Williams	Adams	2J8CNF2KQ
Carol James	Adams	VRSF5JQWZ
David Fuentes	Jefferson	MW5B2VA7Y
Ellen Chu	Jefferson	8VPPS2L39

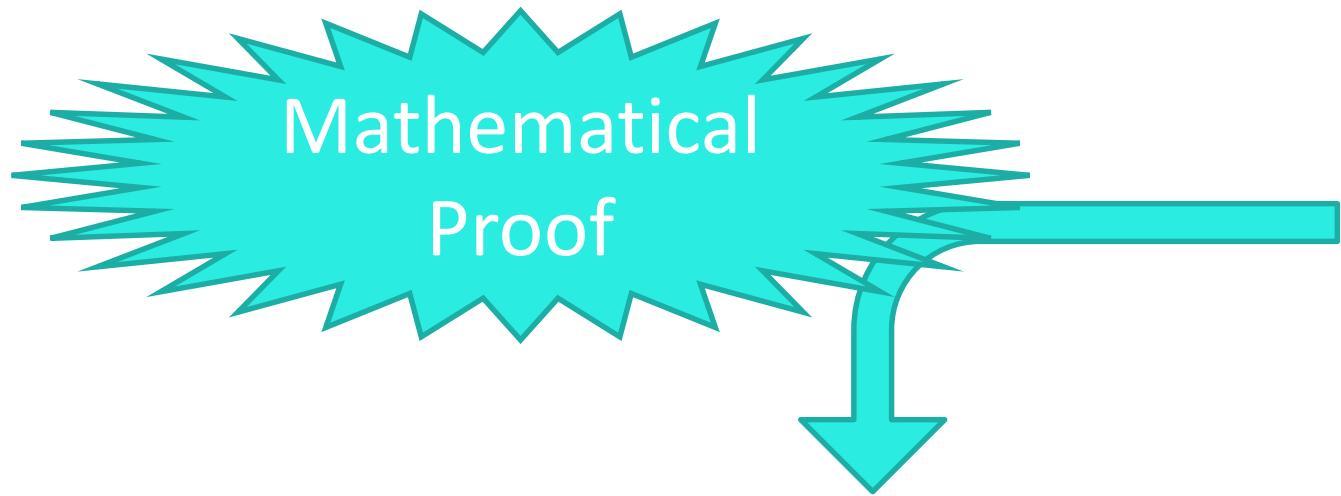
Totals	
Jefferson	3
Adams	2

An E2E-Verifiable Election

X37BM6YPM
2J8CNF2KQ
VRSF5JQWZ
MW5B2VA7Y
8VPSS2L39

Totals	
Jefferson	3
Adams	2

An E2E-Verifiable Election



X37BM6YPM
2J8CNF2KQ
VRSF5JQWZ
MW5B2VA7Y
8VPPS2L39

Totals	
Jefferson	3
Adams	2

End-to-End Verifiable Elections

Two questions must be answered ...

1. How do voters turn their preferences into encrypted votes?
2. How are voters convinced that the published set of encrypted votes corresponds the announced tally?

The Tally Proof

There are essentially two paradigms to choose from ...

- Anonymized Ballots
- Ballotless Tallying

Anonymized Ballots



Ballotless Tallying



Traditional Static Encryption

The only thing you do with encrypted data

VRSF5JQWZ

is decrypt it.

Computing on Encrypted Data

Some modern encryption methods allows useful computation on encrypted data.

VRSF5JQWZ  MW5B2VA7Y

Homomorphic Encryption

We can construct encryption functions such that if

A is *an* encryption of a and

B is *an* encryption of b then

$A \times B$ is *an* encryption of $a \times b$.

Homomorphic Encryption

We can also construct other encryption functions such that if

A is *an* encryption of a and

B is *an* encryption of b then

$A \times B$ is *an* encryption of $a + b$.

Homomorphic Encryption

With RSA encryption,

$$\begin{aligned} Z_1 &= E(M_1) = M_1^e \\ Z_2 &= E(M_2) = M_2^e \end{aligned}$$

$$\begin{aligned} Z_1 \times Z_2 &= E(M_1) \times E(M_2) = M_1^e \times M_2^e \\ &= (M_1 \times M_2)^e = E(M_1 \times M_2) \end{aligned}$$

RSA is *multiplicatively homomorphic*.

Homomorphic Encryption

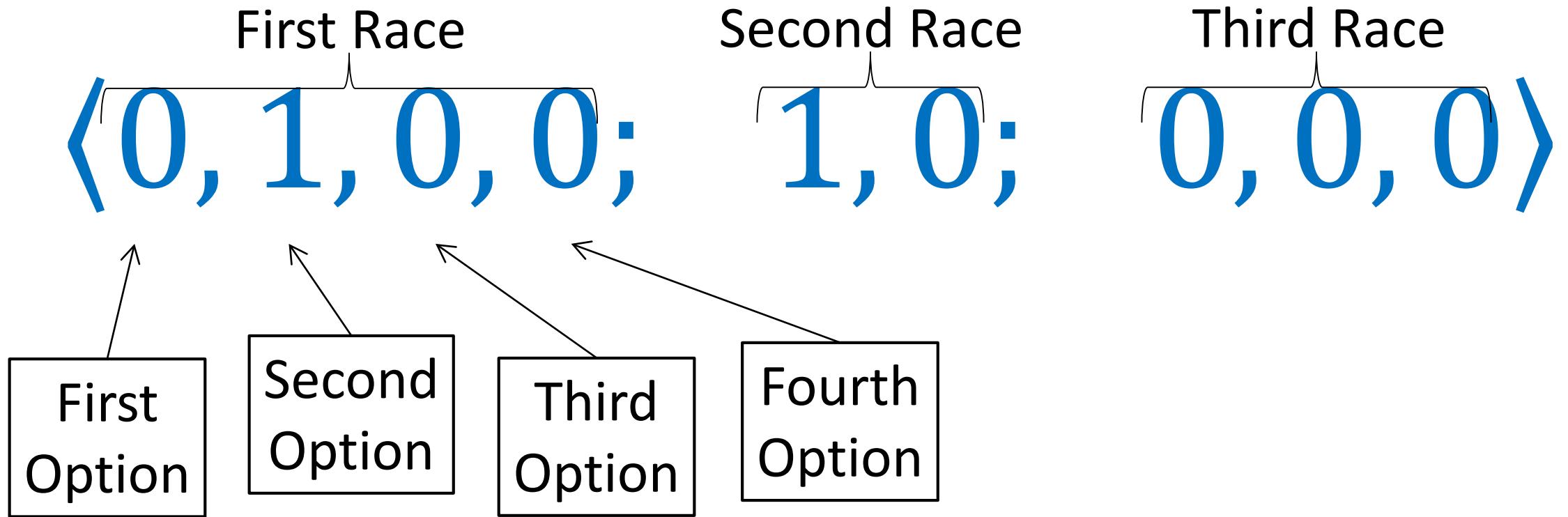
With some other encryption functions,

$$\begin{aligned} Z_1 &= E(M_1) = g^{M_1} \\ Z_2 &= E(M_2) = g^{M_2} \end{aligned}$$

$$\begin{aligned} Z_1 \times Z_2 &= E(M_1) \times E(M_2) = g^{M_1} \times g^{M_2} \\ &= g^{M_1+M_2} = E(M_1 + M_2) \end{aligned}$$

Such functions are *additively homomorphic*.

A Valid Vote



In Elections ...

$$Z_1 = E(\text{Vote \#1})$$

$$Z_2 = E(\text{Vote \#2})$$

⋮

$$Z_k = E(\text{Vote \#}k)$$

The *product* of the *encryptions* of the votes is an *encryption* of the *sum* of the votes.

Homomorphic Encryption

Some Homomorphic Functions

- RSA: $E(M) = M^e \text{ mod } n$
- ElGamal: $E(M, r) = (g^r, Mh^r) \text{ mod } p$
- Goldwasser-Micali: $E(b, r) = r^2g^b \text{ mod } n$
- Benaloh: $E(M, r) = r^e g^M \text{ mod } n$
- Pallier: $E(M, r) = r^n g^M \text{ mod } n^2$

Multiplicative → Additive

RSA and ElGamal are multiplicatively homomorphic.

- To “additively” encrypt message m , compute $M = g^m \text{ mod } n$ and encrypt M .
- Then $M_1 \times M_2 = g^{M_1} \times g^{M_2} = g^{M_1+M_2} \text{ (mod } n\text{)}.$
- Recovering $M_1 + M_2$ requires computing a discrete log, but the plaintext space is small.

Homomorphic Elections

Alice	$\langle 0, 1, 0, 0; \quad 1, 0; \quad 0, 0, 0 \rangle$
Bob	$\langle 0, 0, 0, 1; \quad 1, 0; \quad 0, 1, 0 \rangle$
Carol	$\langle 0, 0, 1, 0; \quad 0, 1; \quad 1, 0, 0 \rangle$
David	$\langle 0, 1, 0, 0; \quad 1, 0; \quad 0, 0, 1 \rangle$
Eve	$\langle 0, 0, 1, 0; \quad 0, 1; \quad 0, 0, 1 \rangle$

Homomorphic Elections

Alice	$\langle 0, 1, 0, 0; \quad 1, 0; \quad 0, 0, 0 \rangle$
Bob	$\langle 0, 0, 0, 1; \quad 1, 0; \quad 0, 1, 0 \rangle$
Carol	$\langle 0, 0, 1, 0; \quad 0, 1; \quad 1, 0, 0 \rangle$
David	$\langle 0, 1, 0, 0; \quad 1, 0; \quad 0, 0, 1 \rangle$
Eve	$\langle 0, 0, 1, 0; \quad 0, 1; \quad 0, 0, 1 \rangle$
$\Sigma =$	$\langle 0, 2, 2, 1; \quad 3, 2; \quad 1, 1, 2 \rangle$

Homomorphic Elections

Alice	$\langle 0, 1, 0, 0; \quad 1, 0; \quad 0, 0, 0 \rangle$
Bob	$\langle 0, 0, 0, 1; \quad 1, 0; \quad 0, 1, 0 \rangle$
Carol	$\langle 0, 0, 1, 0; \quad 0, 1; \quad 1, 0, 0 \rangle$
David	$\langle 0, 1, 0, 0; \quad 1, 0; \quad 0, 0, 1 \rangle$
Eve	$\langle 0, 0, 1, 0; \quad 0, 1; \quad 0, 0, 1 \rangle$

Homomorphic Elections

Alice	$\langle 0, 1, 0, 0; \quad 1, 0; \quad 0, 0, 0 \rangle$
Bob	$\langle 0, 0, 0, 1; \quad 1, 0; \quad 0, 1, 0 \rangle$
Carol	$\langle 0, 0, 1, 0; \quad 0, 1; \quad 1, 0, 0 \rangle$
David	$\langle 0, 1, 0, 0; \quad 1, 0; \quad 0, 0, 1 \rangle$
Eve	$\langle 0, 0, 1, 0; \quad 0, 1; \quad 0, 0, 1 \rangle$

Homomorphic Elections

Alice	$\langle 0, 1, 0, 0; \quad 1, 0; \quad 0, 0, 0 \rangle$
Bob	$\langle 0, 0, 0, 1; \quad 1, 0; \quad 0, 1, 0 \rangle$
Carol	$\langle 0, 0, 1, 0; \quad 0, 1; \quad 1, 0, 0 \rangle$
David	$\langle 0, 1, 0, 0; \quad 1, 0; \quad 0, 0, 1 \rangle$
Eve	$\langle 0, 0, 1, 0; \quad 0, 1; \quad 0, 0, 1 \rangle$
$\Pi =$	$\langle 0, 2, 2, 1; \quad 3, 2; \quad 1, 1, 2 \rangle$

Homomorphic Elections

Alice	$\langle 0, 1, 0, 0; \quad 1, 0; \quad 0, 0, 0 \rangle$
Bob	$\langle 0, 0, 0, 1; \quad 1, 0; \quad 0, 1, 0 \rangle$
Carol	$\langle 0, 0, 1, 0; \quad 0, 1; \quad 1, 0, 0 \rangle$
David	$\langle 0, 1, 0, 0; \quad 1, 0; \quad 0, 0, 1 \rangle$
Eve	$\langle 0, 0, 1, 0; \quad 0, 1; \quad 0, 0, 1 \rangle$
$\Sigma =$	$\langle 0, 2, 2, 1; \quad 3, 2; \quad 1, 1, 2 \rangle$

Homomorphic Elections

Alice	$\langle 0, 1, 0, 0; \quad 1, 0; \quad 0, 0, 0 \rangle$
Bob	$\langle 0, 0, 0, 1; \quad 1, 0; \quad 0, 1, 0 \rangle$
Carol	$\langle 0, 0, 1, 0; \quad 0, 1; \quad 1, 0, 0 \rangle$
David	$\langle 0, 1, 0, 0; \quad 1, 0; \quad 0, 0, 1 \rangle$
Eve	$\langle 0, 0, 1, 0; \quad 0, 1; \quad 0, 0, 1 \rangle$
$\Sigma =$	$\langle 0, 2, 2, 1; \quad 3, 2; \quad 1, 1, 2 \rangle$

Homomorphic Elections

Alice	$\langle 0, 1, 0, 0; \quad 1, 0; \quad 0, 0, 0 \rangle$
Bob	$\langle 0, 0, 0, 1; \quad 1, 0; \quad 0, 1, 0 \rangle$
Carol	$\langle 0, 0, 1, 0; \quad 0, 1; \quad 1, 0, 0 \rangle$
David	$\langle 0, 1, 0, 0; \quad 1, 0; \quad 0, 0, 1 \rangle$
Eve	$\langle 0, 0, 1, 0; \quad 0, 1; \quad 0, 0, 1 \rangle$
$\Sigma =$	$\langle 0, 2, 2, 1; \quad 3, 2; \quad 1, 1, 2 \rangle$

End-to-End Verifiable Elections

Two questions must be answered ...

1. How do voters turn their preferences into encrypted votes?
2. How are voters convinced that the published set of encrypted votes corresponds the announced tally?

An E2E-Verifiable Election

Voter Name	Vote	
Alice Smith	Jefferson	X37BM6YPM
Bob Williams	Adams	2J8CNF2KQ
Carol James	Adams	VRSF5JQWZ
David Fuentes	Jefferson	MW5B2VA7Y
Ellen Chu	Jefferson	8VPPS2L39

Totals	
Jefferson	3
Adams	2

How do Humans Encrypt?

- If voters encrypt their votes with devices of their own choosing, they are subject to coercion and compromise.
- If voters encrypt their votes on “official” devices, how can they trust that their intentions have been properly captured?

The Human Encryptor

We need to find ways to engage humans in an *interactive proof* process to ensure that their intentions are accurately reflected in ballots encrypted on their behalf.

How Can Humans Verify Votes?

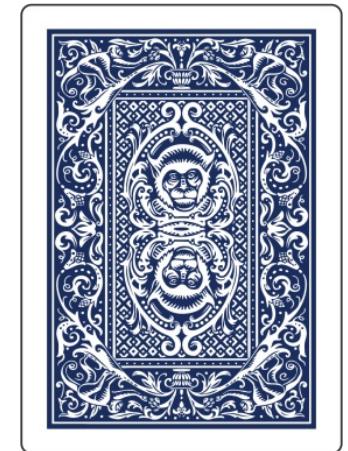
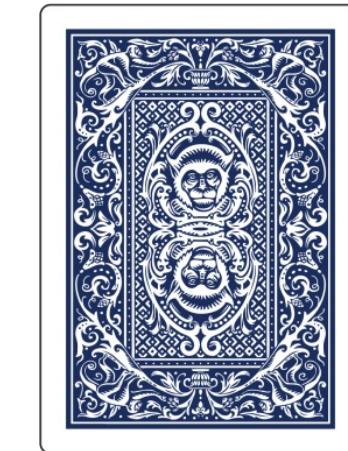
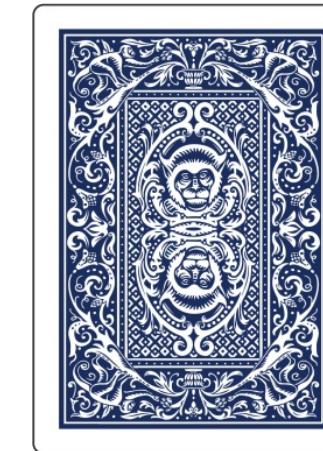
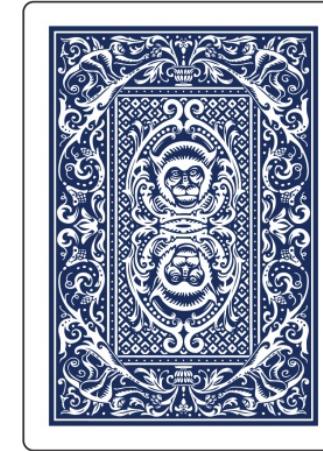
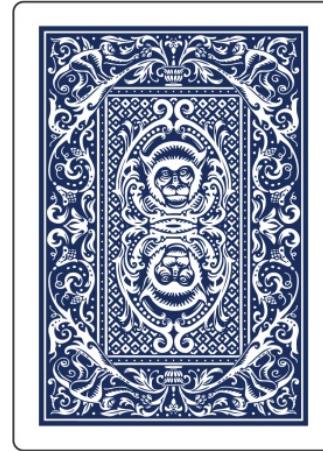
VRSF5JQWZ = Adams ?



Believing Without Seeing

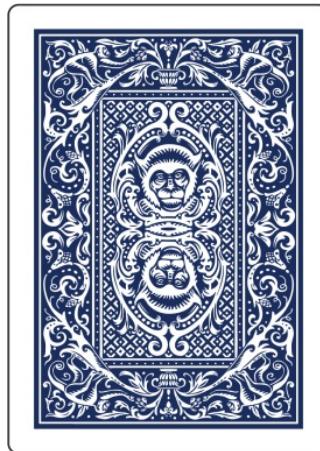
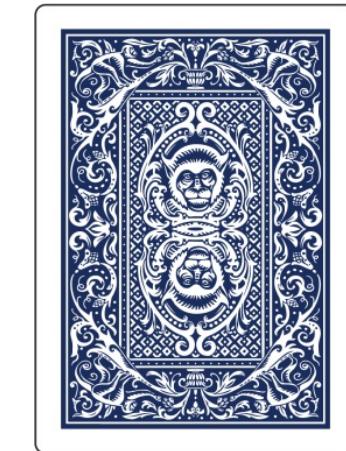
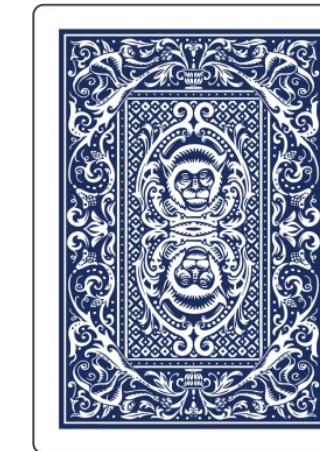
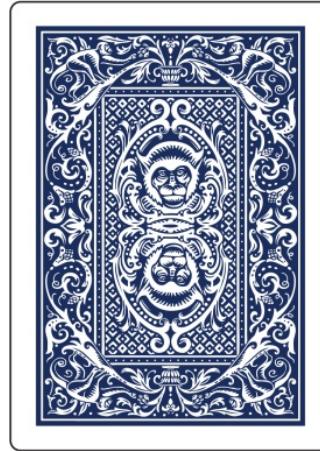
Believing Without Seeing

I claim that all of the cards below are red.



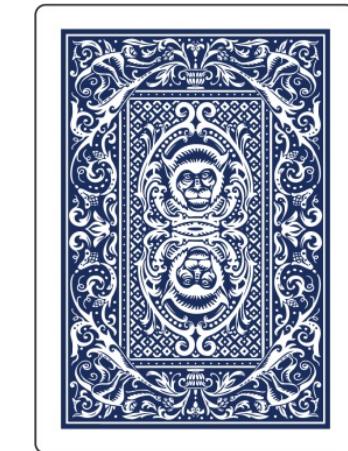
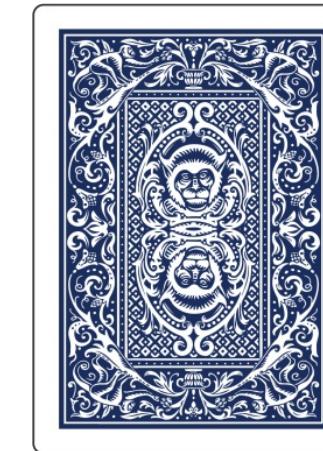
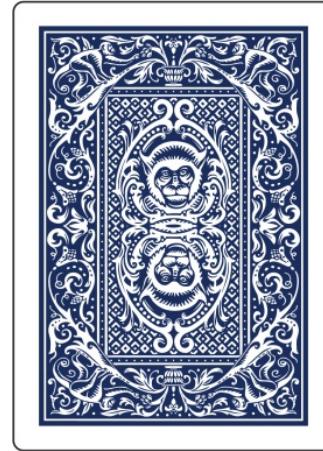
Believing Without Seeing

I claim that all of the cards below are red.



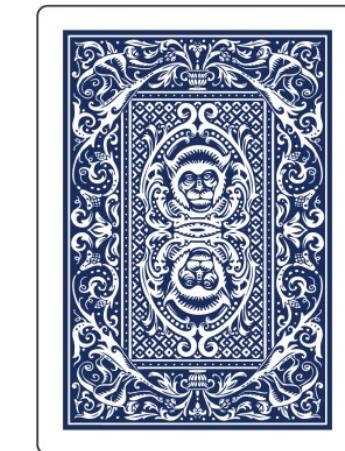
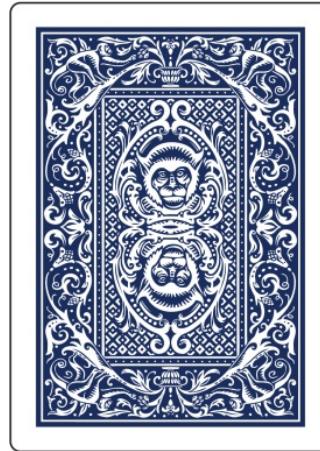
Believing Without Seeing

I claim that all of the cards below are red.



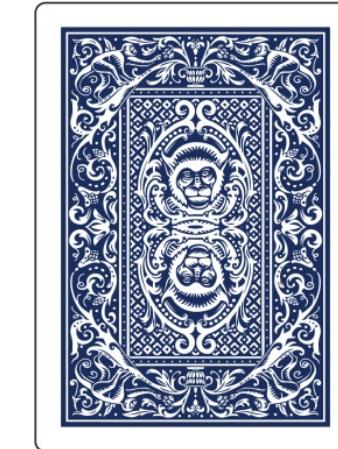
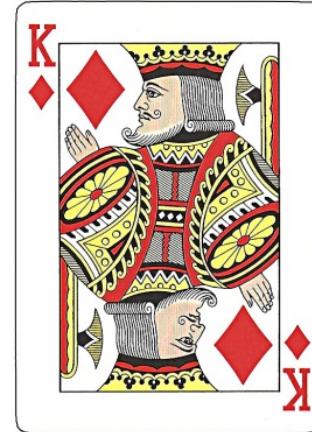
Believing Without Seeing

I claim that all of the cards below are red.



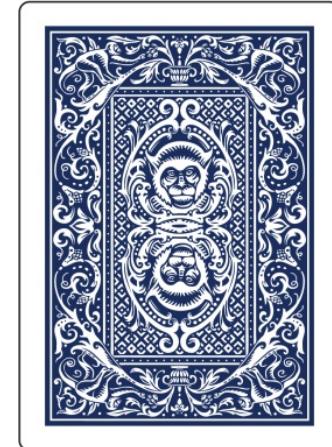
Believing Without Seeing

I claim that all of the cards below are red.



Believing Without Seeing

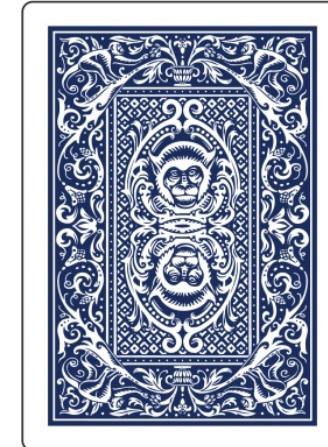
I claim that all of the cards below are red.



Believing Without Seeing

I claim that all of the cards below are red.

You've never seen this card.

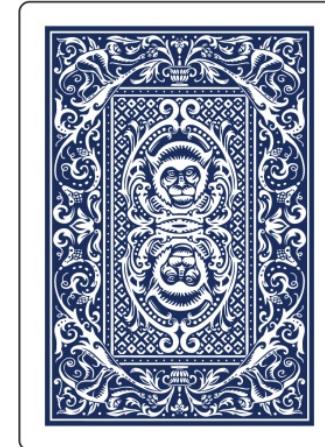


Believing Without Seeing

I claim that all of the cards below are red.

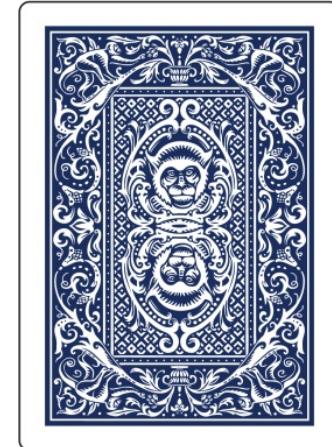
You've never seen this card.

But you now have good
reason to believe it's red.



Non-transferable Belief

Even though you now believe that this card is red, there's nothing that you can do to convince someone else.



Believing Without Seeing

I claim that all of the encryptions below are votes for Adams.

8QZ
4TY
2B7

GX3
9M6
P4Y

T9V
BS5
ZDF

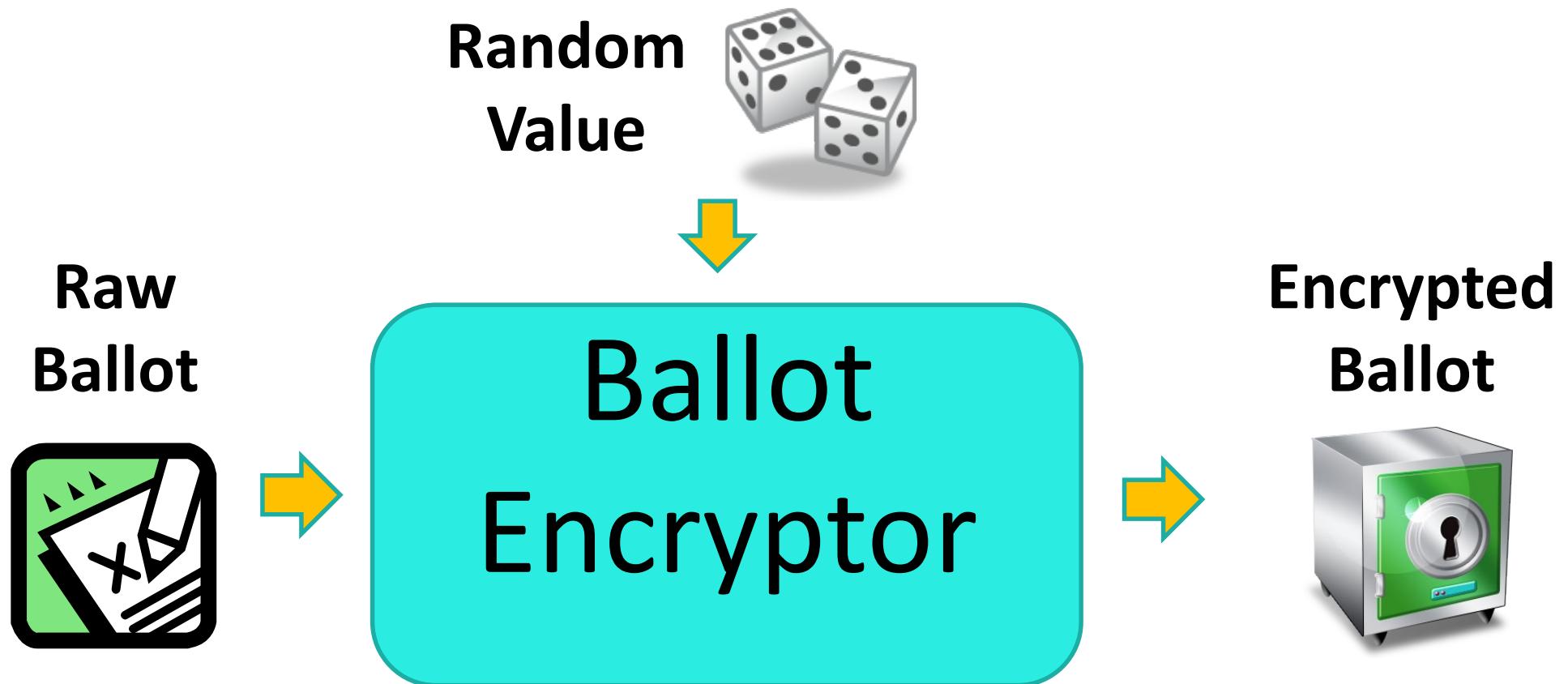
VRS
F5J
QWZ

J44
Y0C
URV

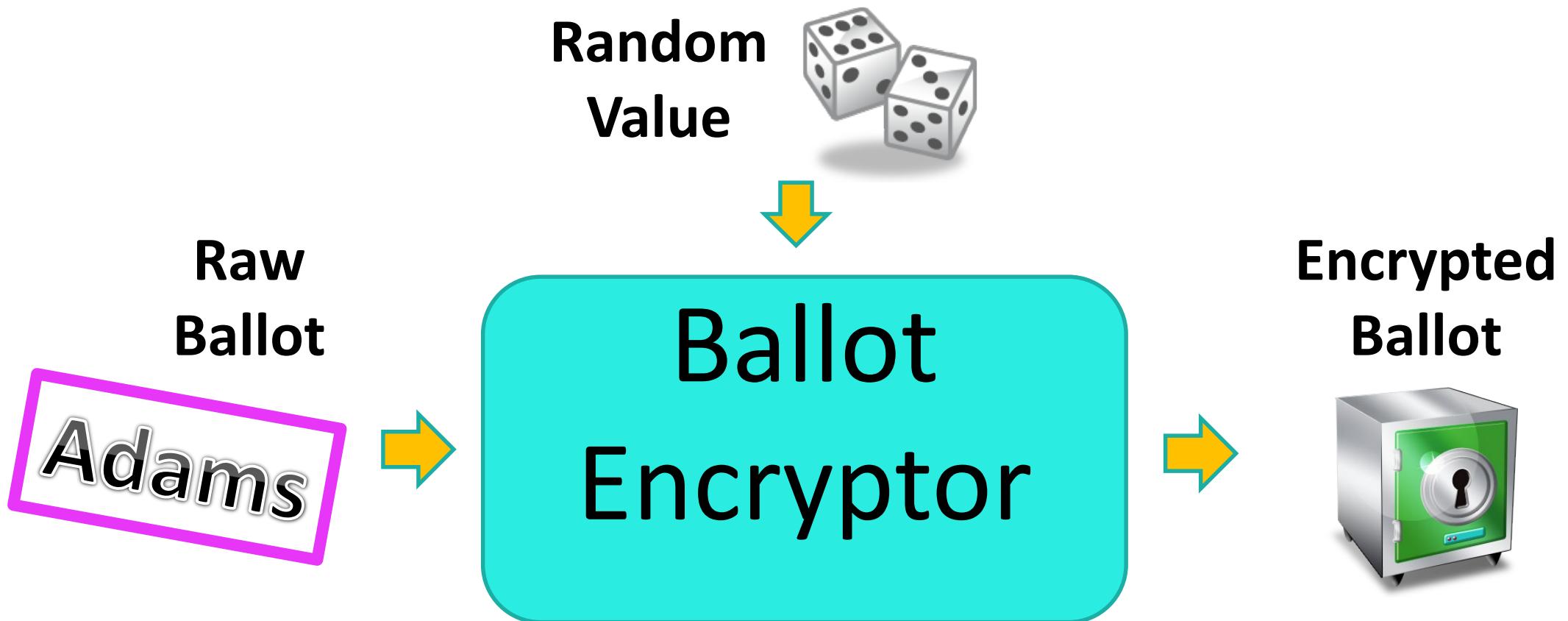
Randomized Encryption

- Ballot encryption *must* be “randomized”.
- Identical ballots should *not* have identical encryptions.

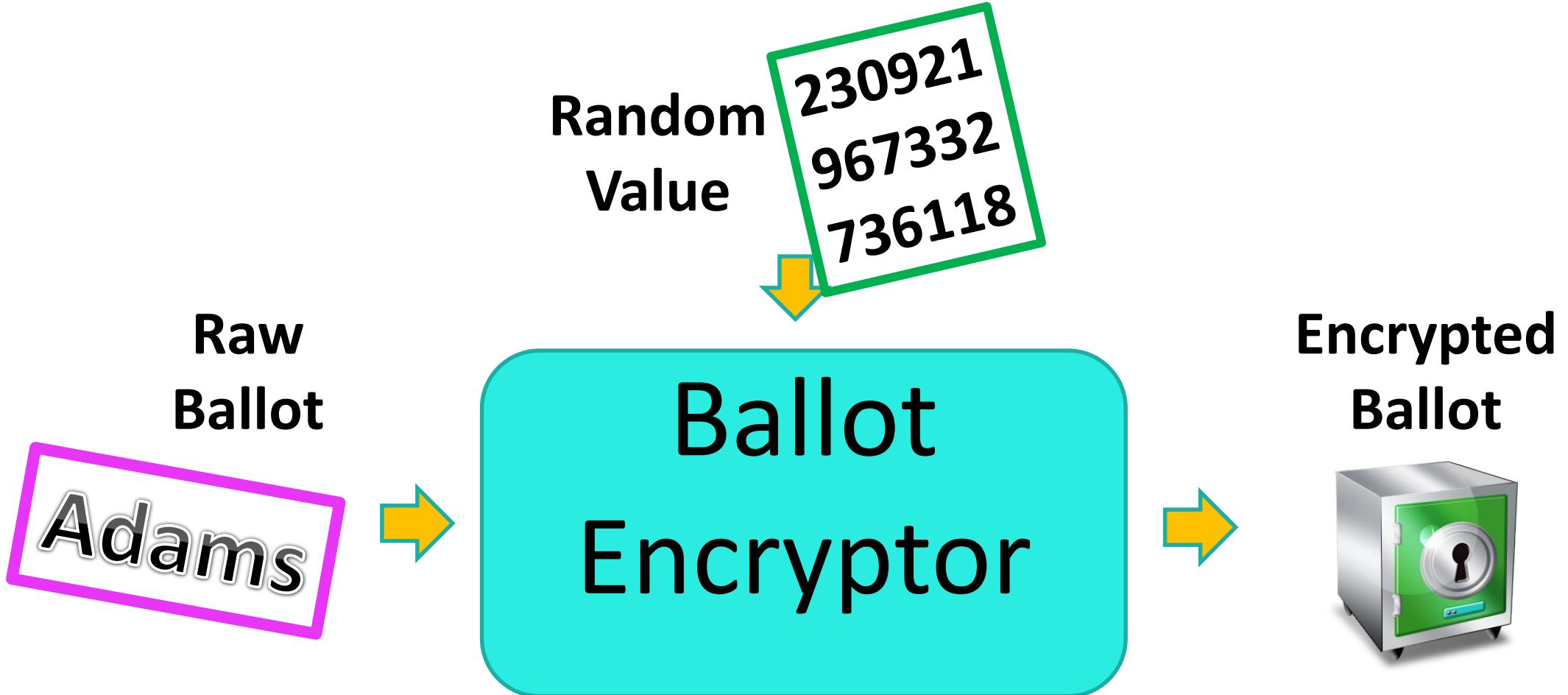
Ballot Encryption



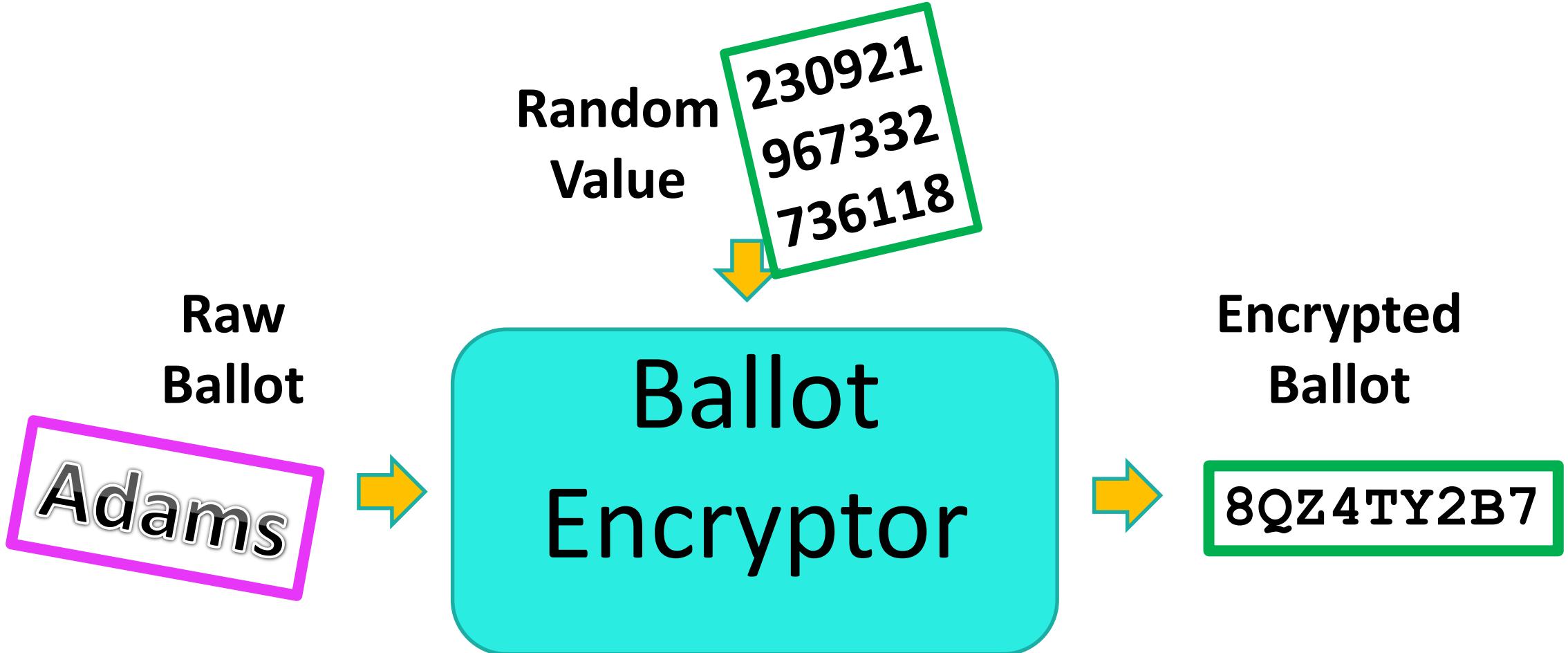
Ballot Encryption



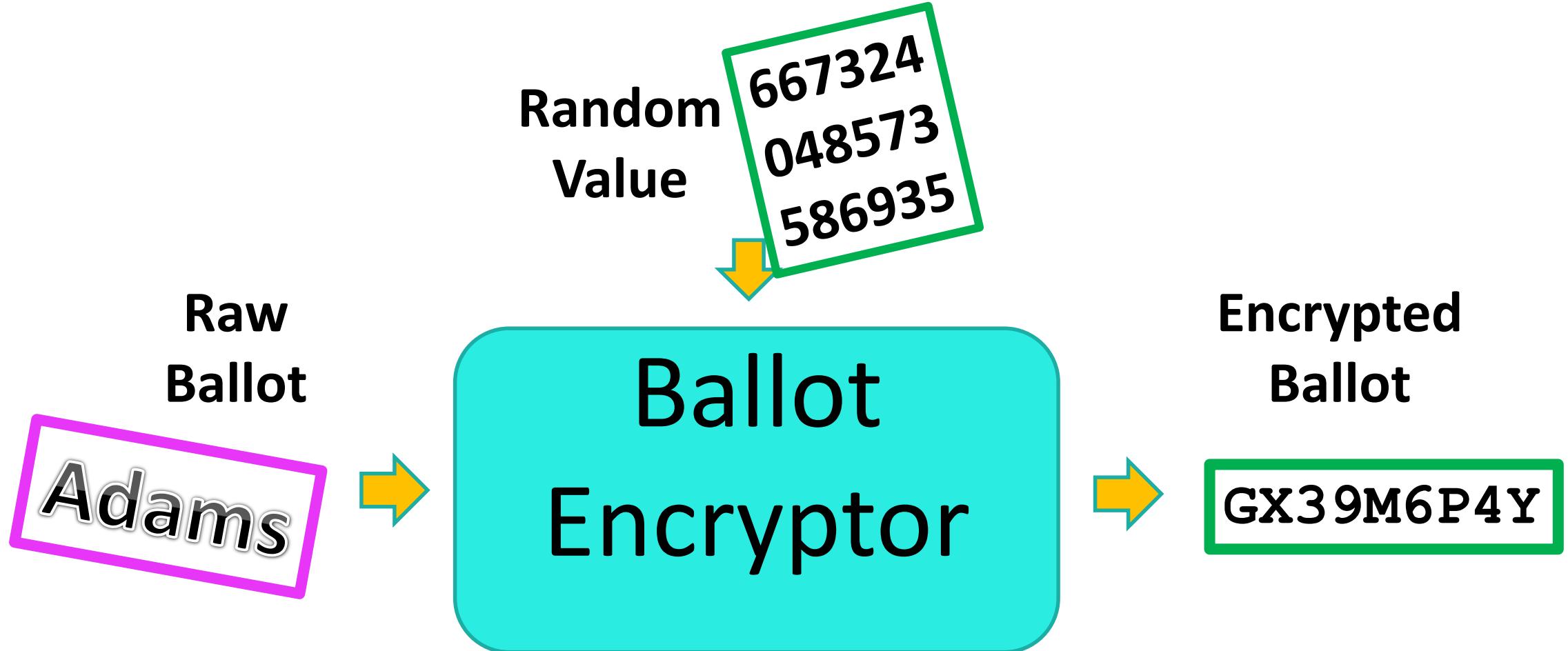
Ballot Encryption



Ballot Encryption



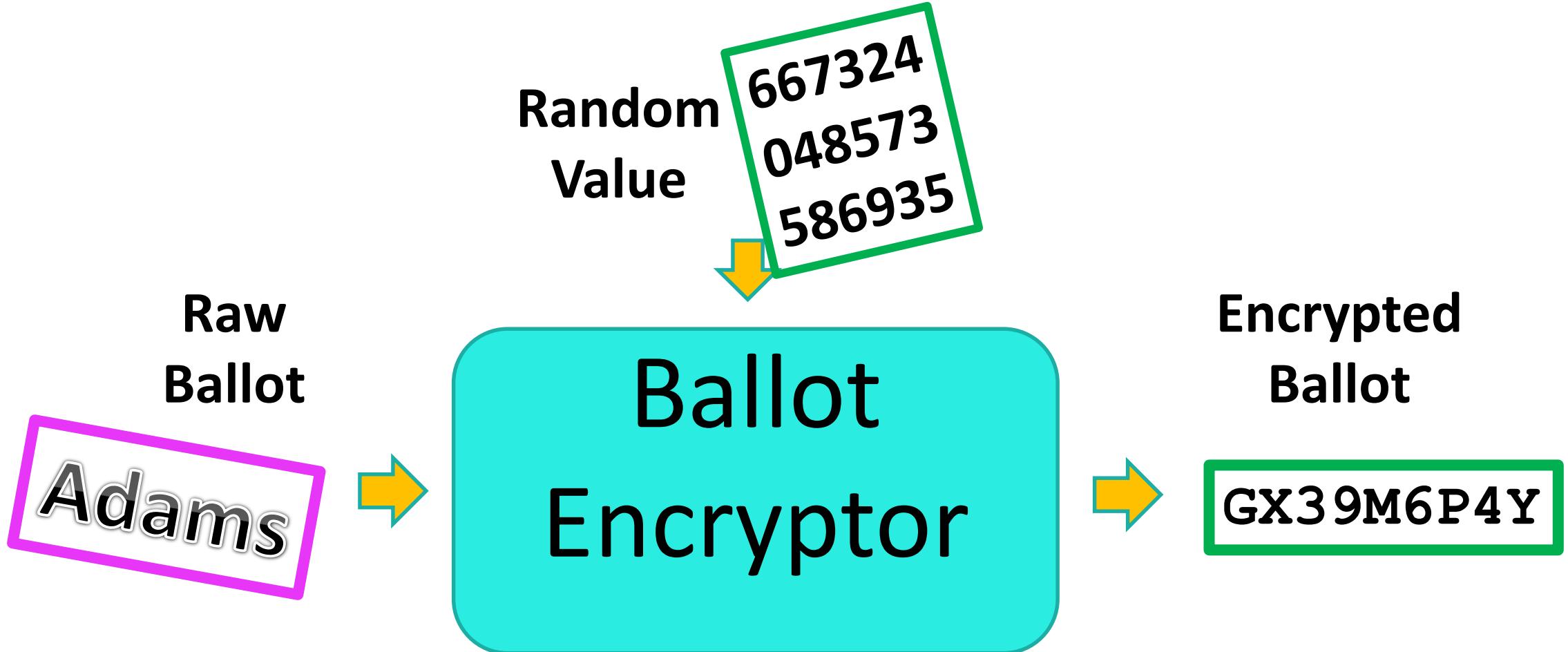
Ballot Encryption



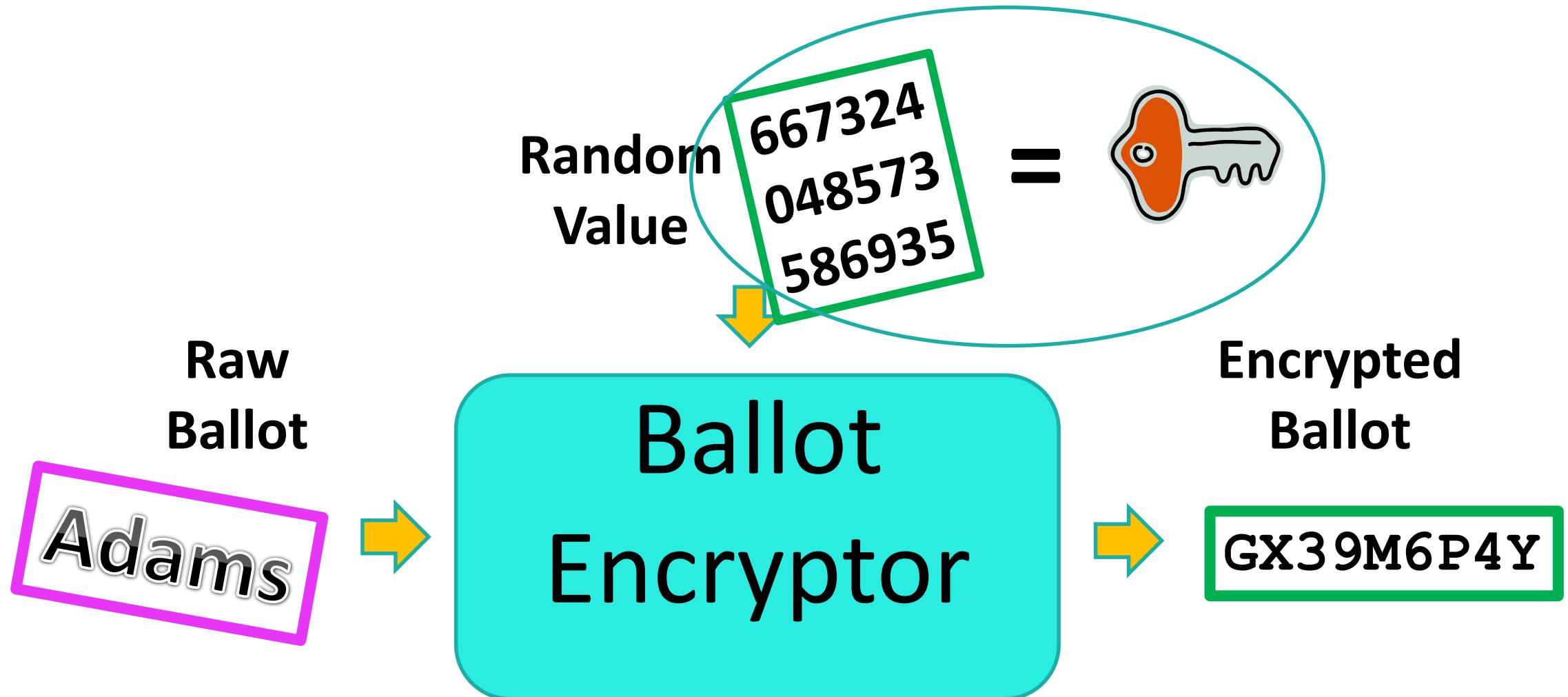
Ballot Decryption

One can enable an object to be *verifiably decrypted* by revealing the random value used to encrypt it.

Ballot Encryption



Ballot Encryption



Believing Without Seeing

I claim that all of the encryptions below are votes for Adams.

8QZ
4TY
2B7

GX3
9M6
P4Y

T9V
BS5
ZDF

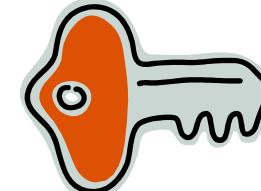
VRS
F5J
QWZ

J44
Y0C
URV

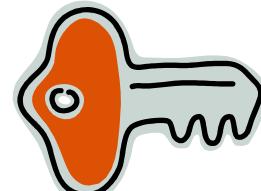
Believing Without Seeing

I claim that all of the encryptions below are votes for Adams.

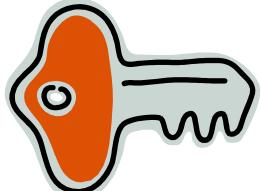
8QZ
4TY
2B7



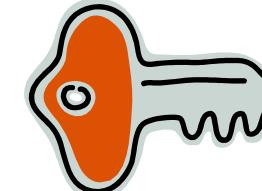
GX3
9M6
P4Y



T9V
BS5
ZDF



VRS
F5J
QWZ



J44
Y0C
URV

Believing Without Seeing

I claim that all of the encryptions below are votes for Adams.



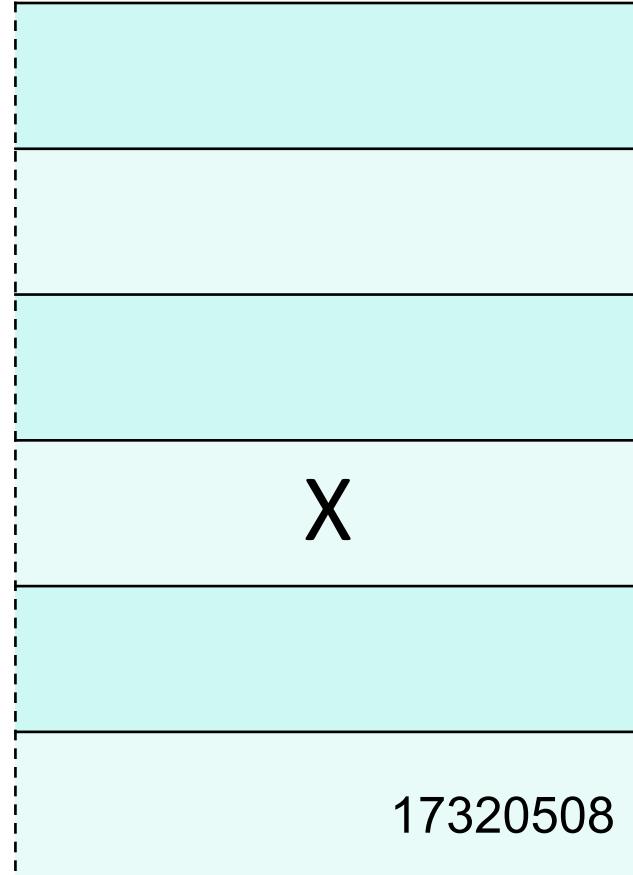
Prêt à Voter Ballot

Bob	
Eve	
Carol	
Alice	
David	
	17320508

Prêt à Voter Ballot

Bob	
Eve	
Carol	
Alice	X
David	
	17320508

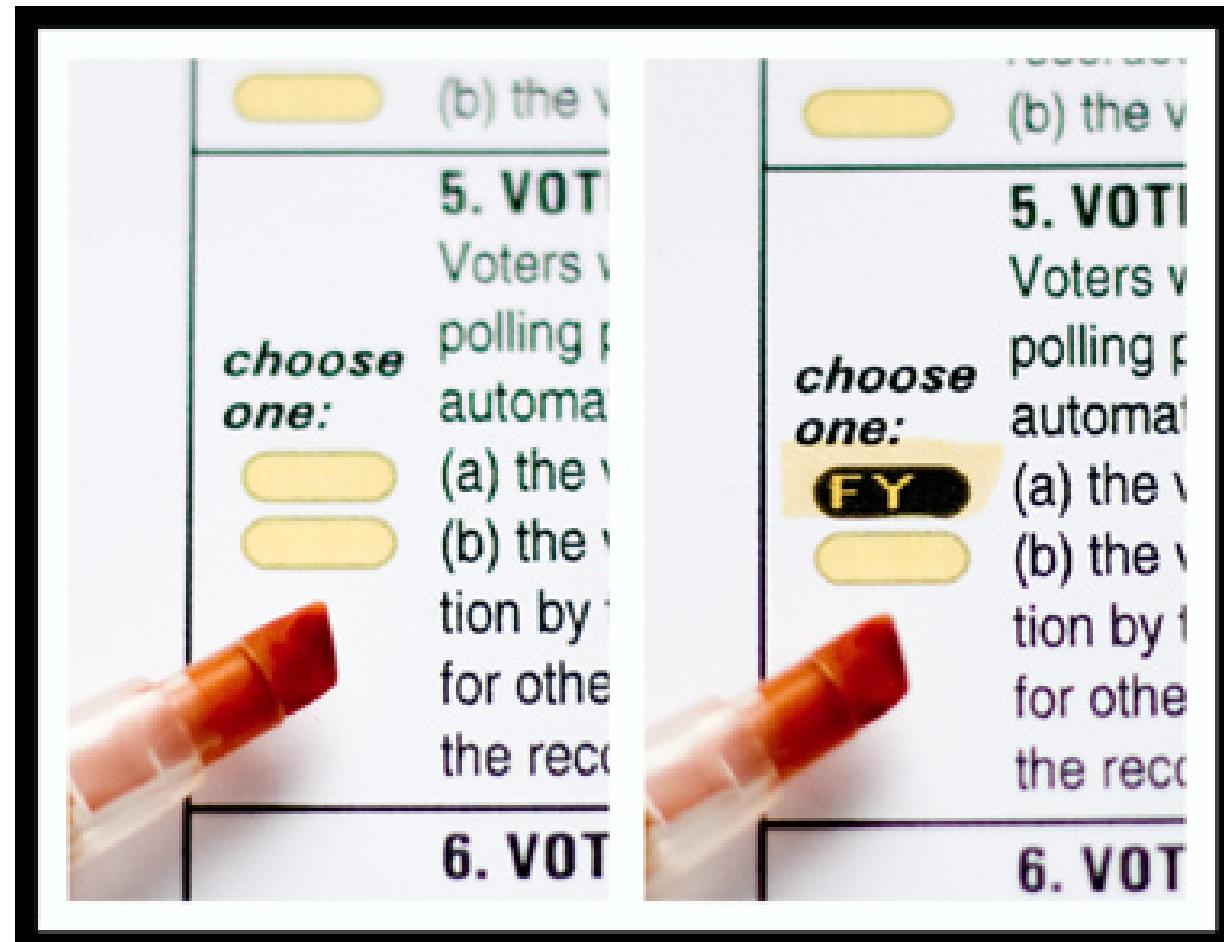
Prêt à Voter Ballot



Prêt à Voter Ballot Verification

- Many more ballots are produced than needed.
- Pre-election auditing tests decrypt many ballots.
- Voters may select and *spoil* additional ballots.
- Leftover ballots are audited post-election.

Scantegrity



Scantegrity Ballot Verification

Similar to Prêt à Voter

Voter-Initiated Auditing

- Voter can use “any” device to make selections (touch-screen DRE, OpScan, etc.)
- After selections are made, voter receives an encrypted receipt of the ballot.

Voter-Initiated Auditing



Voter choice: Cast or Spoil

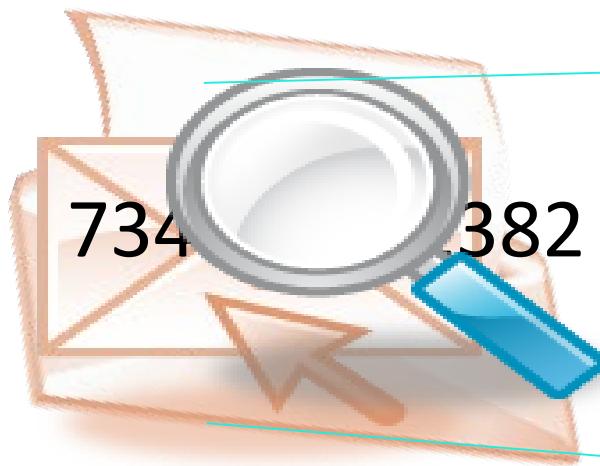
Voter-Initiated Auditing

Cast



Voter-Initiated Auditing

Spoil



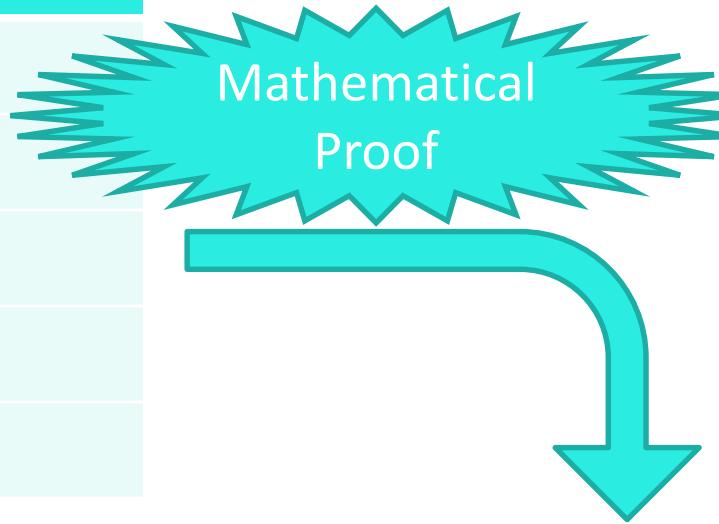
Vote for **Alice**
Random # is
28637582738

Voters Needn't Decrypt

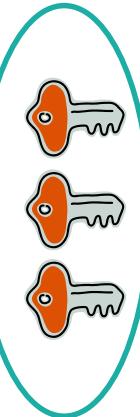
- A public election record includes *both* cast ballots *and* spoiled ballots.
- Diligent voters need do nothing more than to check that their ballots are properly recorded.

A Verifiable Election Record

Cast Ballots
X37BM6YPM
2J8CNF2KQ
VRSF5JQWZ
MW5B2VA7Y
8VPPS2L39



Spoiled Ballots	
36PWY4MMB	Jefferson
8QZ4TY2B7	Adams
GX39M6P4Y	Adams



Totals	
Jefferson	3
Adams	2

In practice ...

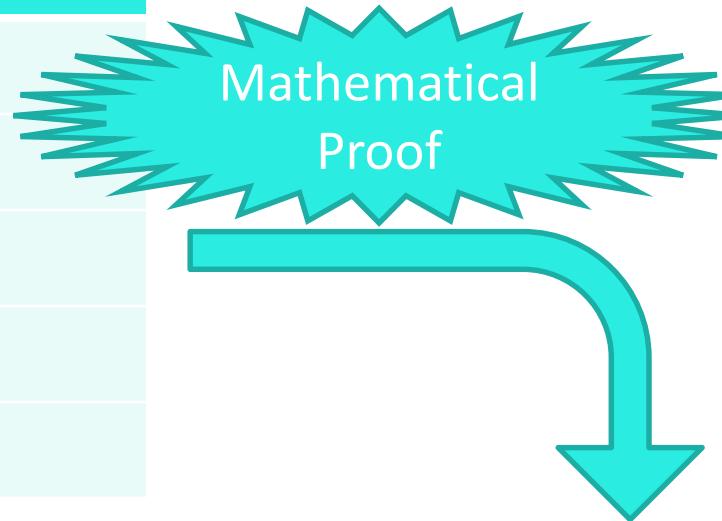
- Even if very few voters each “spoil” a single ballot, very high integrity is assured.
- If 100 voters in a national election each spoil a single ballot, a malicious system would be unlikely to be able to alter even 1% of the votes without detection.

Asymmetric Cryptography

- Devices *don't* need to have secret keys!
- They need to be able to *encrypt*, but they needn't be able to *decrypt*.

A Verifiable Election Record

Cast Ballots
X37BM6YPM
2J8CNF2KQ
VRSF5JQWZ
MW5B2VA7Y
8VPPS2L39



Spoiled Ballots	
36PWY4MMB	Jefferson
8QZ4TY2B7	Adams
GX39M6P4Y	Adams

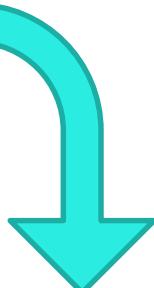
Totals	
Jefferson	3
Adams	2

A Verifiable Election Record

Cast Ballots

X37BM6YPM
2J8CNF2KQ
VRSF5JQWZ
MW5B2VA7Y
8VPPS2L39

Mathematical
Proof



Totals

Jefferson	3
Adams	2

A Verifiable Election Record

Cast Ballots

X37BM6YPM

2J8CNF2KQ

VRSF5JQWZ

MW5B2VA7Y

8VPPS2L39

Totals

Jefferson

3

Adams

2

A Verifiable Election Record

Cast Ballots	Adams	Jefferson
X37BM6YPM	0	1
2J8CNF2KQ	1	0
VRSF5JQWZ	1	0
MW5B2VA7Y	0	1
8VPPS2L39	0	1

Totals	
Jefferson	3
Adams	2

A Verifiable Election Record

Cast Ballots	Adams	Jefferson
X37BM6YPM	0	1
2J8CNF2KQ	1	0
VRSF5JQWZ	1	0
MW5B2VA7Y	0	1
8VPPS2L39	0	1

Totals	
Jefferson	3
Adams	2

A Verifiable Election Record

Cast Ballots	Adams	Jefferson
X37BM6YPM	0	1
2J8CNF2KQ	1	0
VRSF5JQWZ	1	0
MW5B2VA7Y	0	1
8VPPS2L39	0	1

Totals	
Jefferson	3
Adams	2

A Verifiable Election Record

Cast Ballots	Adams	Jefferson
X37BM6YPM	0	1
2J8CNF2KQ	1	0
VRSF5JQWZ	1	0
MW5B2VA7Y	0	1
8VPPS2L39	0	1

x

Totals	
Jefferson	3
Adams	2

A Verifiable Election Record

Cast Ballots	Adams	Jefferson
X37BM6YPM	0	1
2J8CNF2KQ	1	0
VRSF5JQWZ	1	0
MW5B2VA7Y	0	1
8VPPS2L39	0	1

X

CM97JQX4D

Totals	
Jefferson	3
Adams	2

A Verifiable Election Record

Cast Ballots	Adams	Jefferson
X37BM6YPM	0	1
2J8CNF2KQ	1	0
VRSF5JQWZ	1	0
MW5B2VA7Y	0	1
8VPPS2L39	0	1

x

CM97JQX4D

+

Totals	
Jefferson	3
Adams	2

A Verifiable Election Record

Cast Ballots	Adams	Jefferson
X37BM6YPM	0	1
2J8CNF2KQ	1	0
VRSF5JQWZ	1	0
MW5B2VA7Y	0	1
8VPPS2L39	0	1

x

+

CM97JQX4D	2	3
-----------	---	---

Totals	
Jefferson	3
Adams	2

A Verifiable Election Record

Cast Ballots	Adams	Jefferson
X37BM6YPM	0	1
2J8CNF2KQ	1	0
VRSF5JQWZ	1	0
MW5B2VA7Y	0	1
8VPPS2L39	0	1

X

+

CM97JQX4D	2	3
-----------	---	---

Totals	
Jefferson	3
Adams	2

A Verifiable Election Record

Cast Ballots	Adams	Jefferson
X37BM6YPM	0	1
2J8CNF2KQ	1	0
VRSF5JQWZ	1	0
MW5B2VA7Y	0	1
8VPPS2L39	0	1

x

+

CM97JQX4D	2	3
-----------	---	---

Totals	
Jefferson	3
Adams	2

A Verifiable Election Record

Cast Ballots	Adams	Jefferson
X37BM6YPM	0	1
2J8CNF2KQ	1	0
VRSF5JQWZ	1	0
MW5B2VA7Y	0	1
8VPPS2L39	0	1
×		+
CM97JQX4D	2	3

Spoiled Ballots	
36PWY4MMB	Jefferson
8QZ4TY2B7	Adams
GX39M6P4Y	Adams

Totals	
Jefferson	3
Adams	2

A Verifiable Election Record

Cast Ballots	Adams	Jefferson
X37BM6YPM	0	1
2J8CNF2KQ	1	0
VRSF5JQWZ	1	0
MW5B2VA7Y	0	1
8VPPS2L39	0	1

X

+

CM97JQX4D	2	3
-----------	---	---

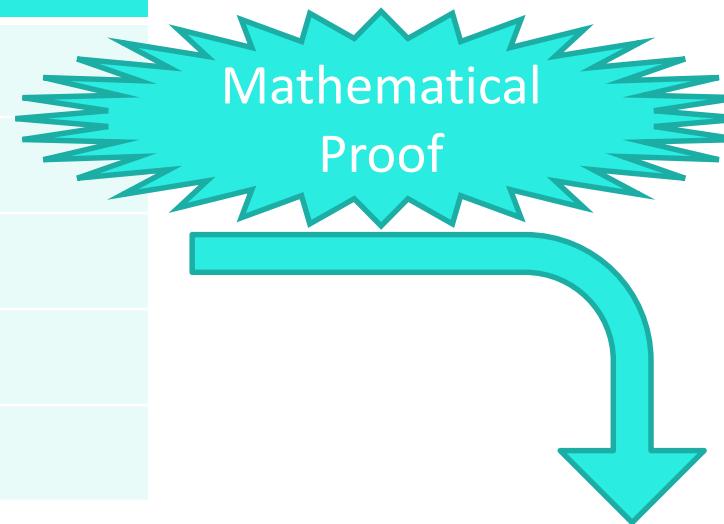
Spoiled Ballots	
36PWY4MMB	Jefferson
8QZ4TY2B7	Adams
GX39M6P4Y	Adams



Totals	
Jefferson	3
Adams	2

A Verifiable Election Record

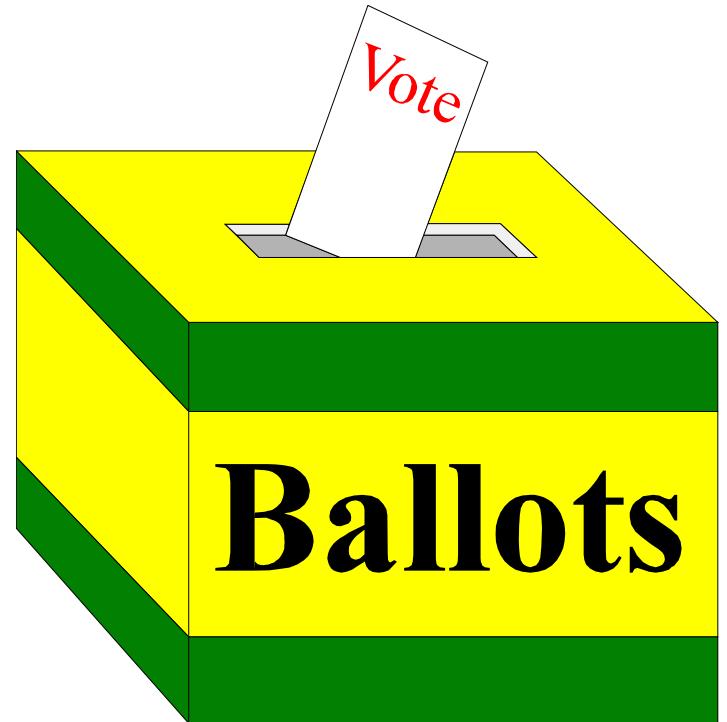
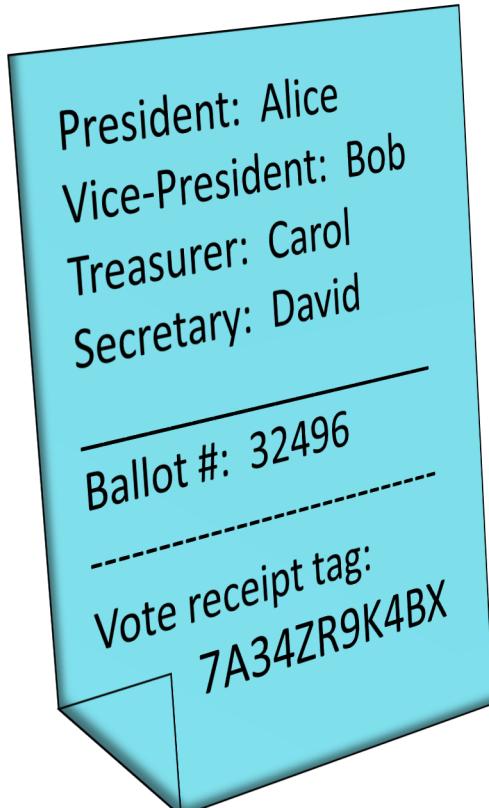
Cast Ballots
X37BM6YPM
2J8CNF2KQ
VRSF5JQWZ
MW5B2VA7Y
8VPPS2L39



Spoiled Ballots	
36PWY4MMB	Jefferson
8QZ4TY2B7	Adams
GX39M6P4Y	Adams

Totals	
Jefferson	3
Adams	2

STAR-Vote

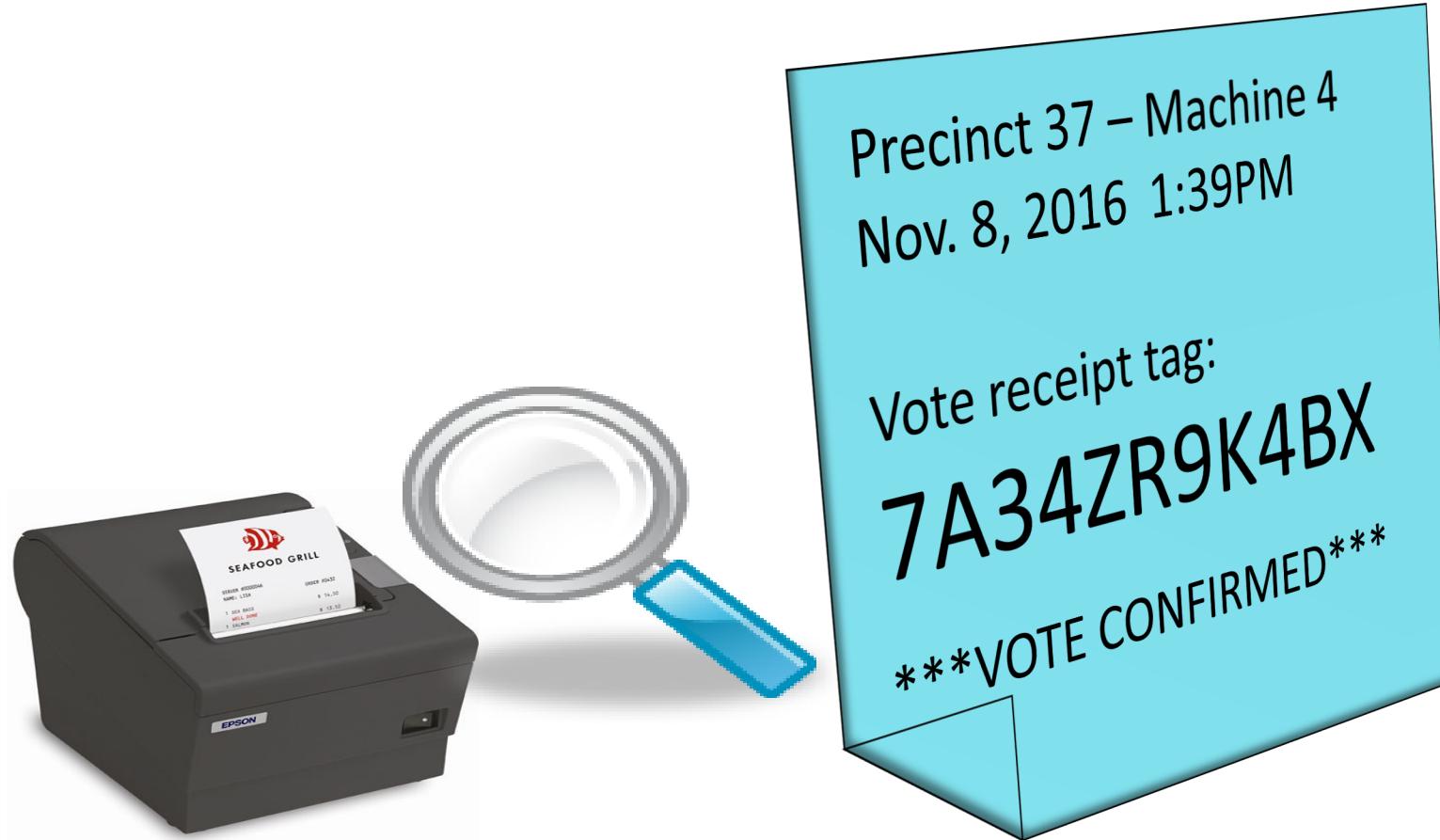


The Voter's Perspective

Verifiable election systems can be built to look exactly like current systems ...

... with one addition ...

A Verifiable Receipt



STAR-Vote Ballot Verification

- Ballots are encrypted by ballot marking device.
- Once voters receive their receipts, they may choose to *cast* or *spoil* their ballots.

The Voter's Perspective

Voters can ...

- Use receipts to check their results are properly recorded on a public web site.
- Throw their receipts in the trash.
- Write and use their own election verifiers.
- Download applications from sources of their choice to verify the mathematical proof of the tally.
- Believe verifications done by their political parties, LWV, ACLU, etc.
- Accept the results without question.

Real-World Deployments

- Helios (www.heliosvoting.org) – Adida and others
 - Used to elect president of UC Louvain, Belgium.
 - Used in Princeton University student government.
 - Used by ACM, IACR, and other professional societies.
- Scantegrity II (www.scantegrity.org) – Chaum, Rivest, many others
 - Used for 2009 & 2011 municipal elections in Takoma Park, MD.
- STAR-Vote – Benaloh, Byrne, Eakin, Kortum, McBurnett, Pereira, Stark, Wallach
 - Designed for use in Travis County, Texas.

The “Great” Rift

There is a long-standing dispute amongst election advocates.

- Most security experts say that paper ballots are necessary for good election security.
- Accessibility advocates say that paper is a non-starter for voters with visual or motor skill needs.

The “Great” Compromise

In a 2017 meeting at the headquarters of the Election Assistance Commission in Silver Springs, it was agreed that compliant voting systems must either

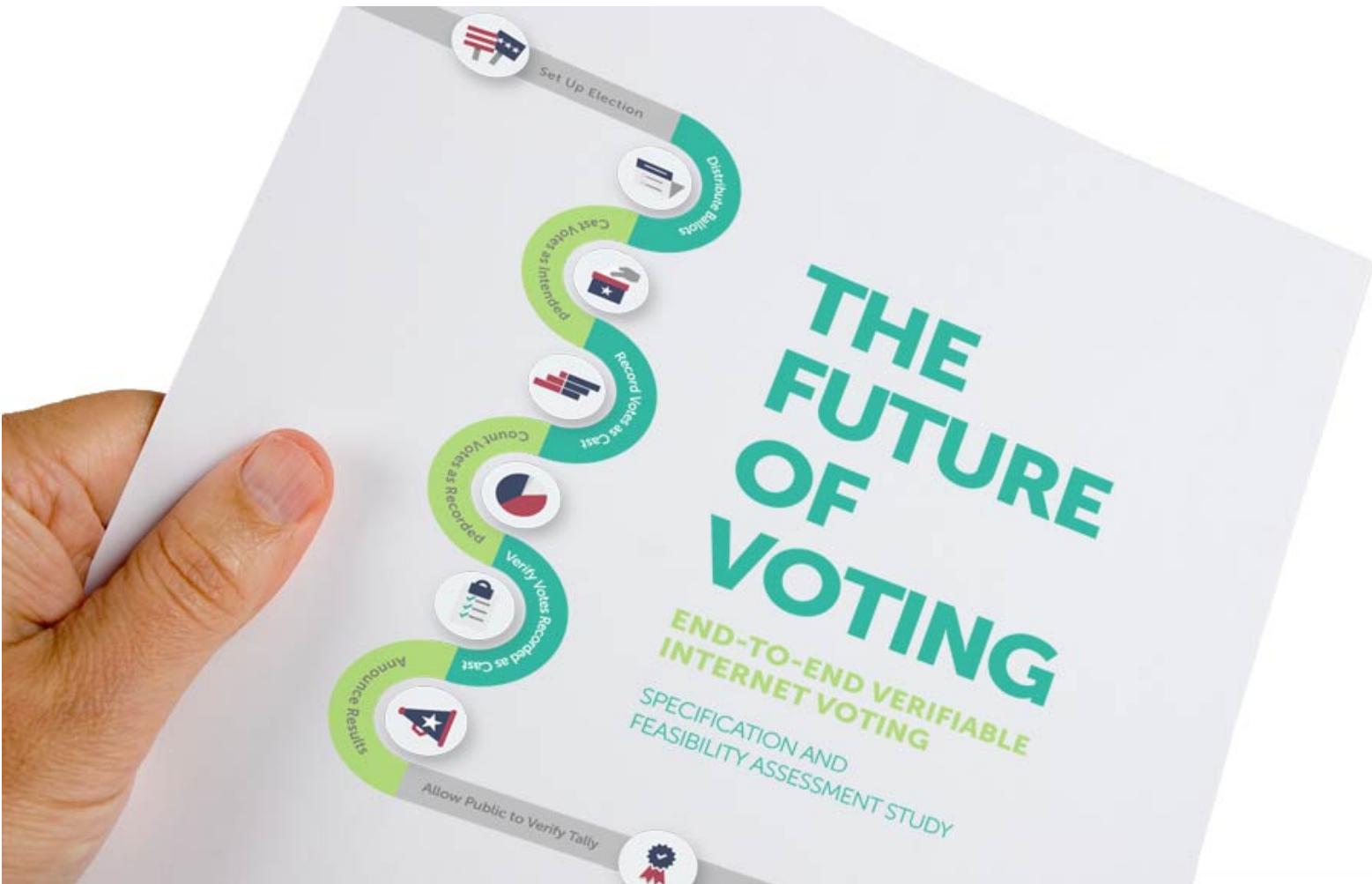
1. be paper-based, or
2. be end-to-end verifiable.

(Note that a voting system *can* be both.)

What's Next?

Internet Voting?

- Some jurisdictions are beginning to explore Internet voting.
- There is a strong push towards IV from a variety of constituencies.



U.S. Vote Foundation E2E-VIV

Principal Conclusions

1. Any public elections conducted over the Internet must be end-to-end verifiable.
2. No Internet voting system of any kind should be used for public elections before end-to-end verifiable in-person voting systems have been widely deployed and experience has been gained from their use.



Questions ???