

Week 6



STEVENS
INSTITUTE *of* TECHNOLOGY
THE INNOVATION UNIVERSITY®



An Introduction to Cyber Security – CS 573

Instructor: Dr. Edward G. Amoroso
eamoroso@tag-cyber.com

Required Week Six Readings

1. “Why Cryptosystems Fail,” Ross Anderson
<https://www.cl.cam.ac.uk/~rja14/Papers/wcf.pdf>
2. Finish Reading “*From CIA to APT: An Introduction*”
to Cyber Security, E. Amoroso & M. Amoroso

Twitter: @hashtag_cyber
LinkedIn: Edward Amoroso



Week 6: Symmetric Cryptography

Reminder: What are the
Three Methods of Cryptanalysis?

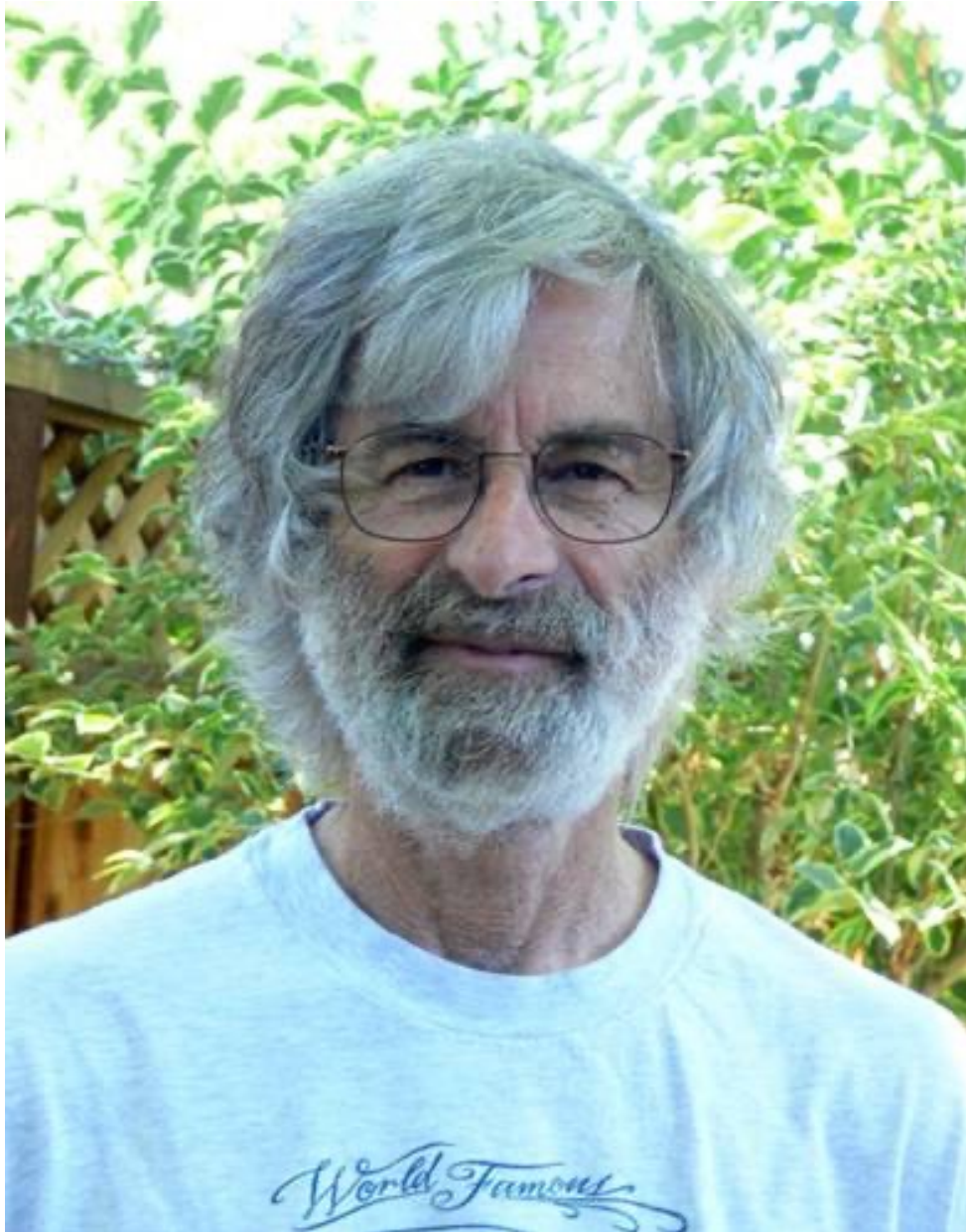
Three Cryptanalytic Methods

Ciphertext-Only: In this method of cryptanalysis, the attacker has access only to a collection of ciphertexts.

Known-Plaintext: In this method of cryptanalysis, the attacker has a set of ciphertexts to which they know the corresponding plaintext.

Chosen-Plaintext: In the codebook method, the attacker can obtain the ciphertexts (plaintexts) corresponding to an arbitrary set of plaintexts (ciphertexts) of their own choosing.

Are Authentication Protocols with No Challenge
Values Always Ciphertext-Only?



Technical Note
Operating Systems

Anita K. Jones
Editor

Password Authentication with Insecure Communication

Leslie Lamport
SRI International

A method of user password authentication is described which is secure even if an intruder can read the system's data, and can tamper with or eavesdrop on the communication between the user and the system. The method assumes a secure one-way encryption function and can be implemented with a microcomputer in the user's terminal.

Key Words and Phrases: security, authentication, passwords, one-way function
CR Categories: 4.35, 4.39

I. The Problem

In remotely accessed computer systems, a user identifies himself to the system by sending a secret password. There are three ways an intruder could learn the user's secret password and then impersonate him when interacting with the system:

- (1) By gaining access to the information stored inside the system, e.g., reading the system's password file.
- (2) By intercepting the user's communication with the system, e.g., eavesdropping on the line connecting the user's terminal with the system, or observing the execution of the password checking program.
- (3) By the user's inadvertent disclosure of his password, e.g., choosing an easily guessed password.

The third possibility cannot be prevented by any password protocol, since two individuals presenting the same password information cannot be distinguished by the system. Eliminating this possibility requires some mechanism for physically identifying the user—for ex-

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

This work was funded in part by the National Science Foundation under Grant No. MCS-7816783.

Author's address: Leslie Lamport, SRI International, 333 Ravenswood Avenue, Menlo Park, CA 94025
© 1981 ACM 0001-0782/81/1100-770 \$00.75.

ample, a voice print. Such a mechanism is beyond the scope of this paper, so we restrict ourselves to the problem of removing the first two weaknesses.

II. The Solution

The first weakness can be eliminated by using a *one-way function* to encode the password. A one-way function is a mapping F from some set of words into itself such that:

- (1) Given a word x , it is easy to compute $F(x)$.
- (2) Given a word y , it is not feasible to compute a word x such that $y = F(x)$.

We will not bother to specify precisely what "easy" and "feasible" mean, so our reasoning will be informal. Note that given $F(x)$, it is always possible to find x by an exhaustive search. We require that such a computation be too costly to be practical. A one-way function F can be constructed from a secure encryption algorithm: one computes $F(x)$ by encrypting a standard word using x as a key [1].

Instead of storing the user's password x , the system stores only the value $y = F(x)$. The user identifies himself by sending x to the system; the system authenticates his identity by computing $F(x)$ and checking that it equals the stored value y . Authentication is easy, since our first assumption about F is that it is easy to compute $F(x)$ from x . Anyone examining the system's permanently stored information can discover only y , and by the second assumption about F it will be infeasible for him to compute a value x such that $y = F(x)$. This is a widely used scheme, and is described in [2] and [3].

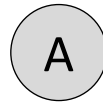
While removing the first weakness, this method does not eliminate the second—an eavesdropper can discover the password x and subsequently impersonate the user. To prevent this, one must use a sequence of passwords $x_1, x_2, \dots, x_{1000}$, where x_i is the password by which the user identifies himself for the i th time. (Of course, the value 1000 is quite arbitrary. The assumption we will tacitly make is that 1000 is small enough so that it is "feasible" to perform 1000 "easy" computations.) The system must know the sequence y_1, \dots, y_{1000} , where $y_i = F(x_i)$, and the y_i must be distinct to prevent an intruder from reusing a prior password.

There are two obvious schemes for choosing the passwords x_i .

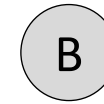
- (1) All the x_i are chosen initially, and the system maintains the entire sequence of values y_1, \dots, y_{1000} in its storage.
- (2) The user sends the value y_{i+1} to the system during the i th session—after logging on with x_i .

Neither scheme is completely satisfactory: the first because both the user and the system must store 1000 pieces of information, and the second because it is not robust—communication failure or interference from an

Lamport S/Key Protocol – Purpose



*A is reporting its
identity to B*



*B is attempting to validate A's reported
identity (i.e., authenticating A)*

Lamport S/Key Protocol – Set-Up

A

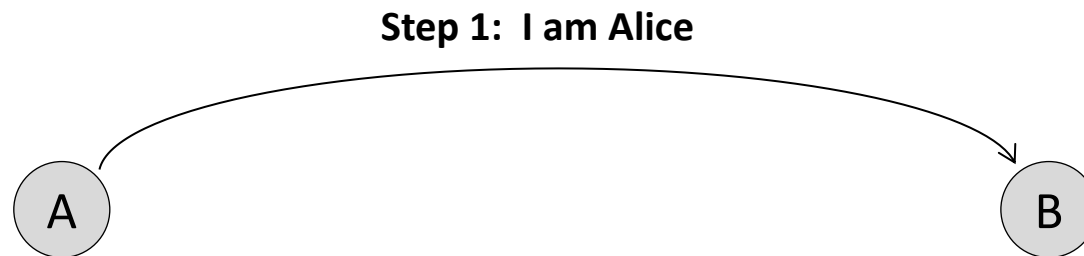
B

*B Does Not Store
The Seed Value (λ)*

Known Function:
f: integer \rightarrow integer
Known Seed:
integer λ
Number of Rounds:
 $n = 10,000$

User	Stored
A	$f, n, f^n(\lambda)$
C	$f', n, f'^n(\lambda')$
G	$f'', n, f''^n(\lambda'')$
...	...

Lamport S/Key Protocol



Known Function:

f : integer \rightarrow integer

Known Seed:

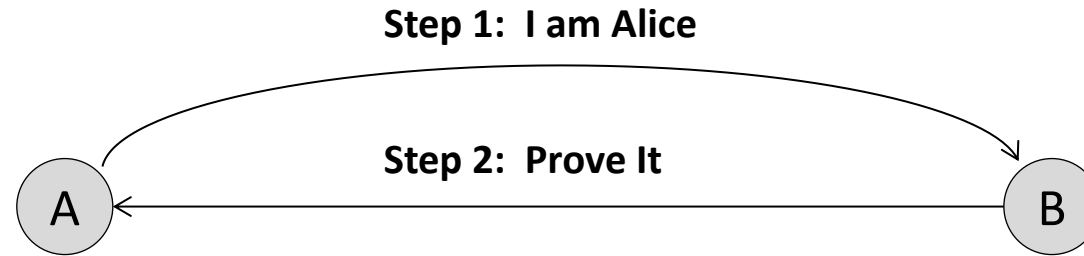
integer λ

Number of Rounds:

$n = 10,000$

<i>User</i>	<i>Stored</i>
A	$f, n, f^n(\lambda)$

Lamport S/Key Protocol



Known Function:

$f: \text{integer} \rightarrow \text{integer}$

Known Seed:

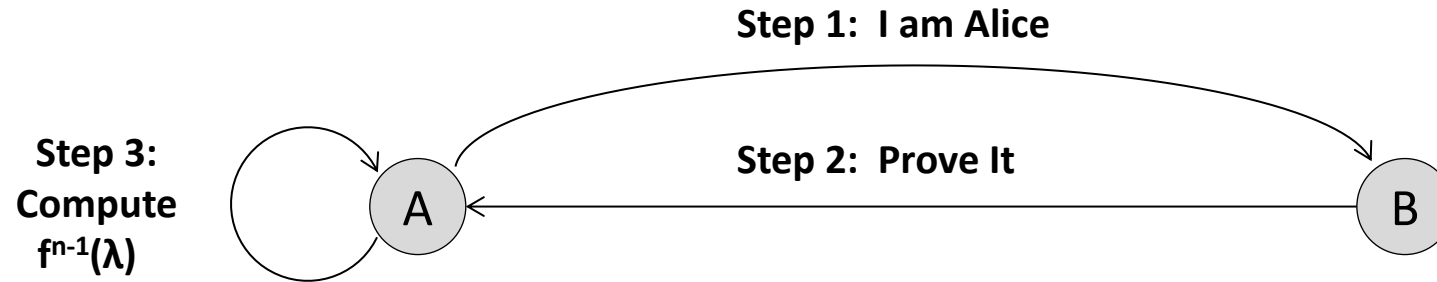
integer λ

Number of Rounds:

$n = 10,000$

<i>User</i>	<i>Stored</i>
A	$f, n, f^n(\lambda)$

Lamport S/Key Protocol



Known Function:

f : integer \rightarrow integer

Known Seed:

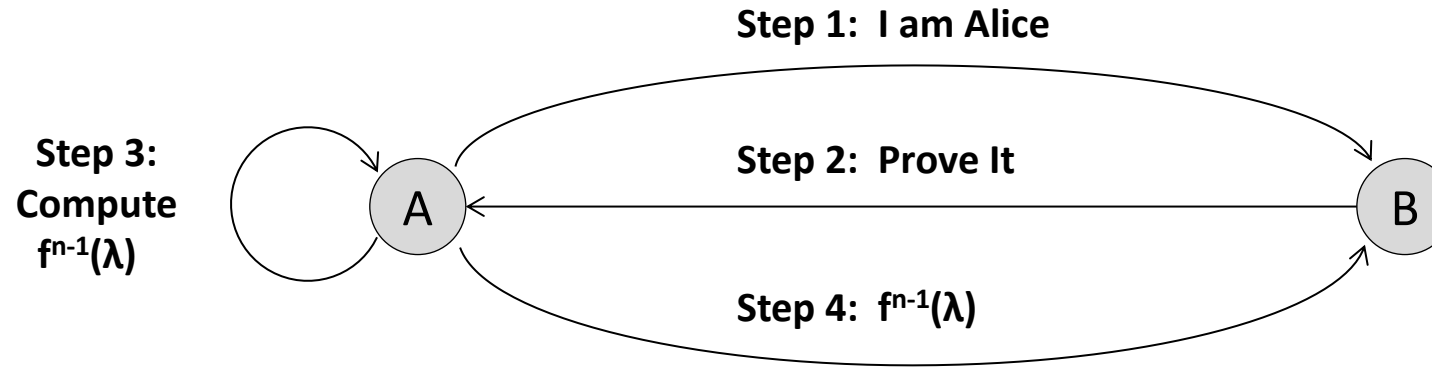
integer λ

Number of Rounds:

$n = 10,000$

<i>User</i>	<i>Stored</i>
A	$f, n, f^n(\lambda)$

Lamport S/Key Protocol



Known Function:

f : integer \rightarrow integer

Known Seed:

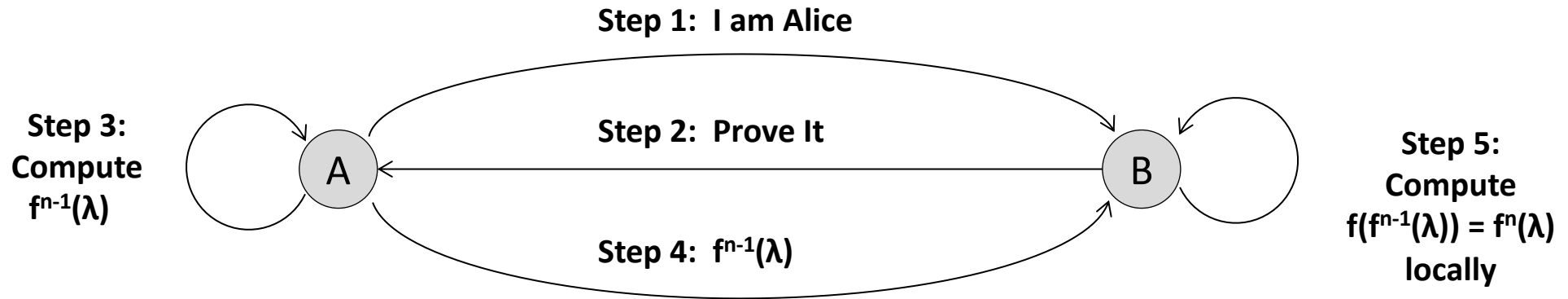
integer λ

Number of Rounds:

$n = 10,000$

<i>User</i>	<i>Stored</i>
A	$f, n, f^n(\lambda)$

Lamport S/Key Protocol



Known Function:

f : integer \rightarrow integer

Known Seed:

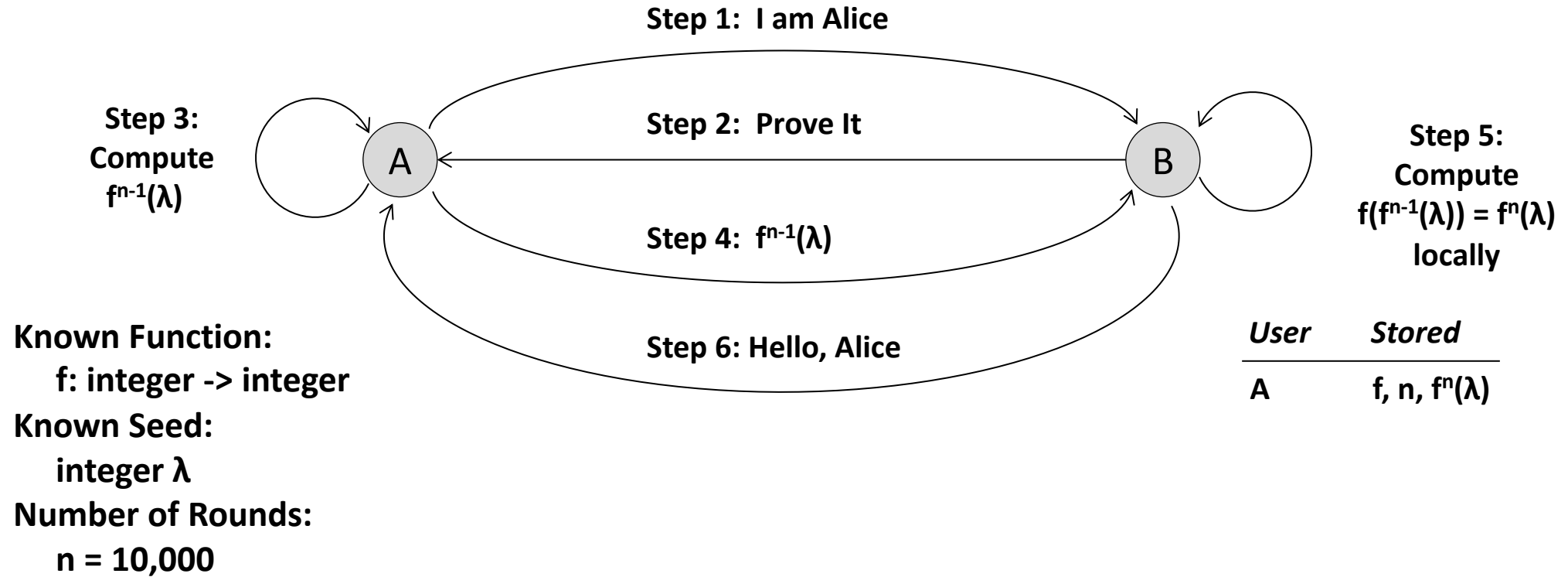
integer λ

Number of Rounds:

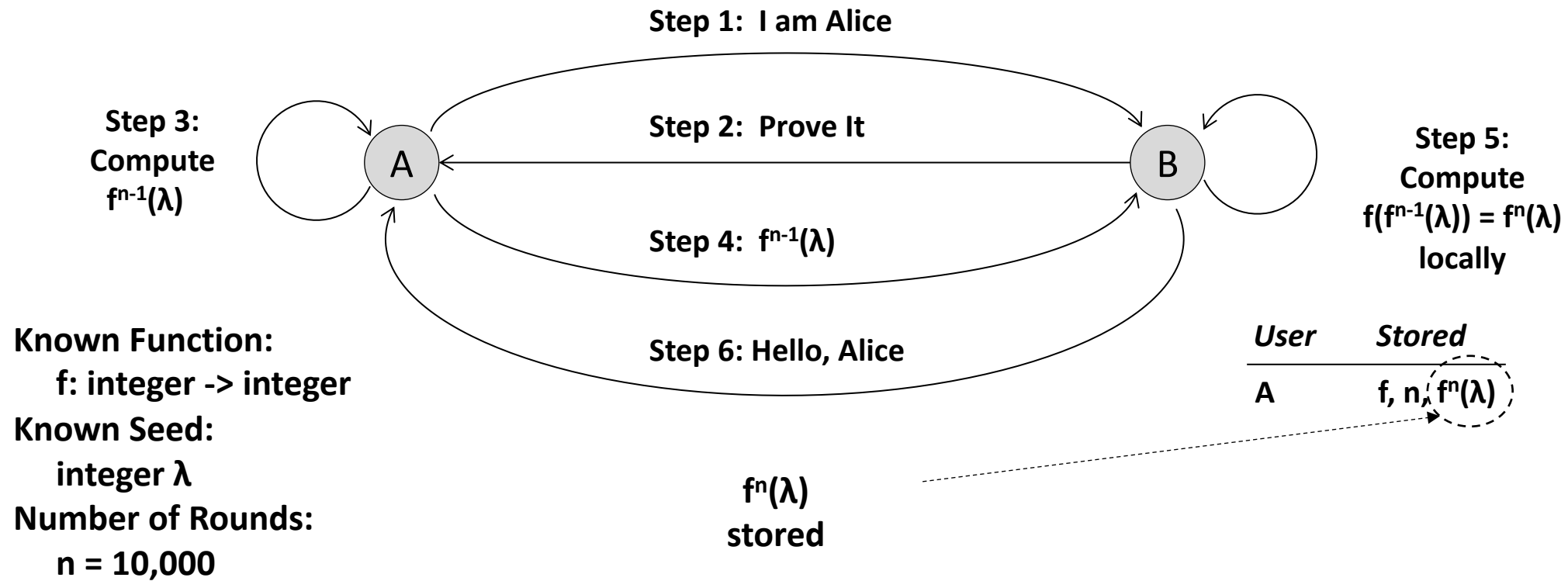
$n = 10,000$

<i>User</i>	<i>Stored</i>
A	$f, n, f^n(\lambda)$

Lamport S/Key Protocol



Lamport S/Key Protocol



Lamport S/Key Protocol

A

B

Known Function:

$f: \text{integer} \rightarrow \text{integer}$

Known Seed:

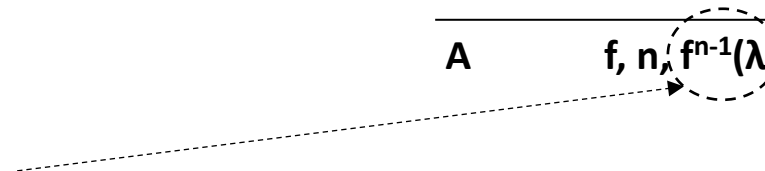
integer λ

Number of Rounds:

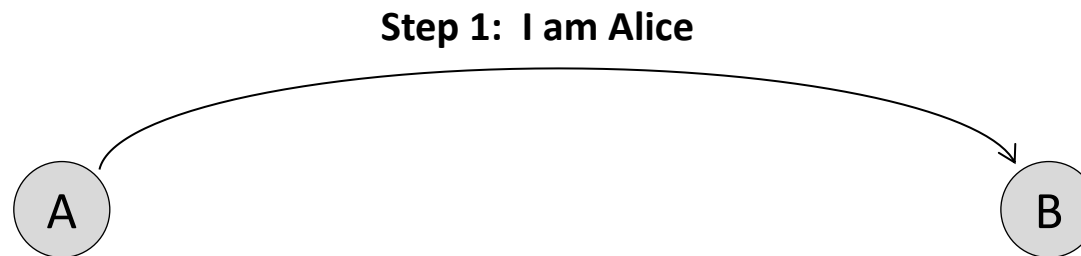
$n-1 = 9,999$

$f^{n-1}(\lambda)$
now stored

User	Stored
A	$f, n, f^{n-1}(\lambda)$



Lamport S/Key Protocol



Known Function:

$f: \text{integer} \rightarrow \text{integer}$

Known Seed:

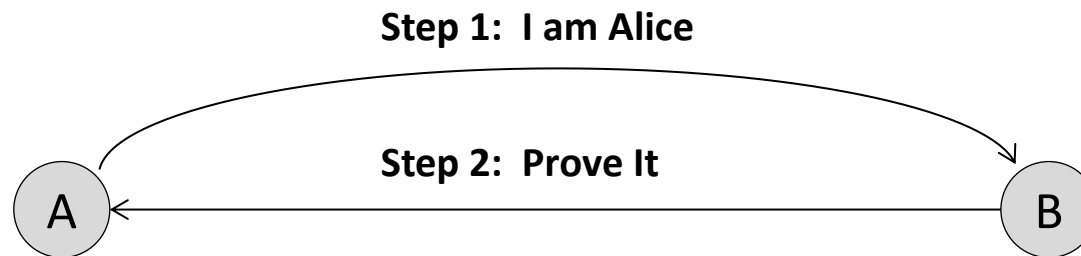
integer λ

Number of Rounds:

$n-1 = 9,999$

<i>User</i>	<i>Stored</i>
A	$f, n, f^{n-1}(\lambda)$

Lamport S/Key Protocol



Known Function:

$f: \text{integer} \rightarrow \text{integer}$

Known Seed:

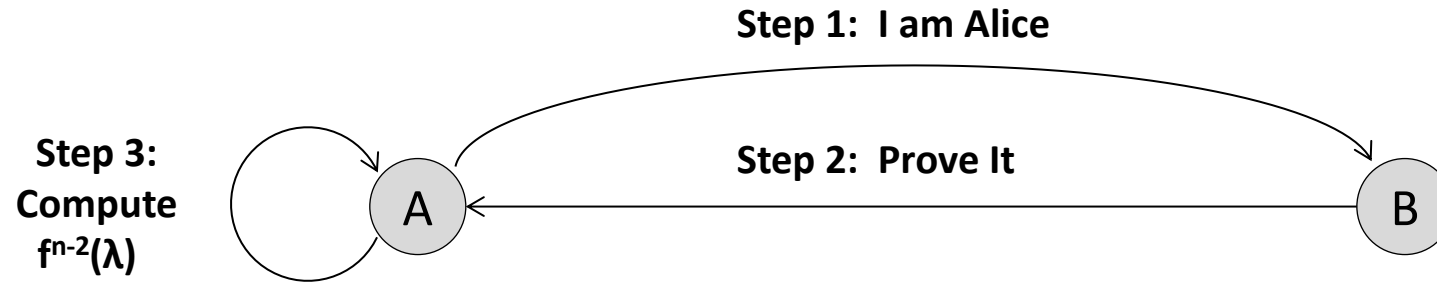
integer λ

Number of Rounds:

$n-1 = 9,999$

<i>User</i>	<i>Stored</i>
A	$f, n, f^{n-1}(\lambda)$

Lamport S/Key Protocol



Known Function:

f: integer \rightarrow integer

Known Seed:

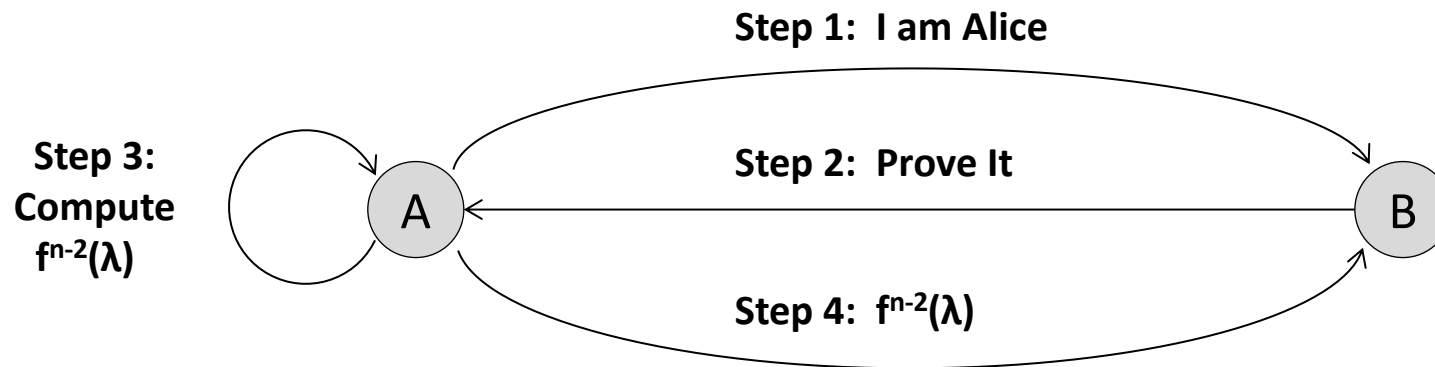
integer λ

Number of Rounds:

$n-1 = 9,999$

<i>User</i>	<i>Stored</i>
A	$f, n, f^{n-1}(\lambda)$

Lamport S/Key Protocol



Known Function:

f : integer \rightarrow integer

Known Seed:

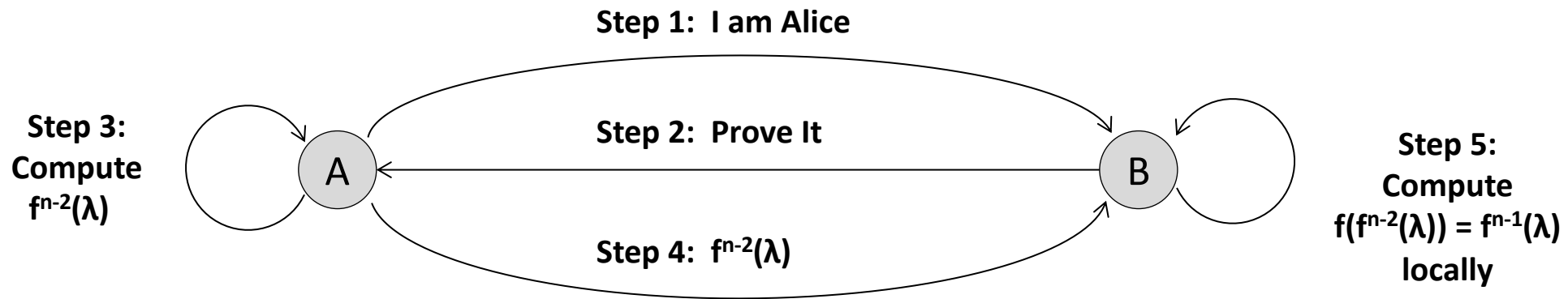
integer λ

Number of Rounds:

$n-1 = 9,999$

<i>User</i>	<i>Stored</i>
A	$f, n, f^{n-1}(\lambda)$

Lamport S/Key Protocol



Known Function:

f : integer \rightarrow integer

Known Seed:

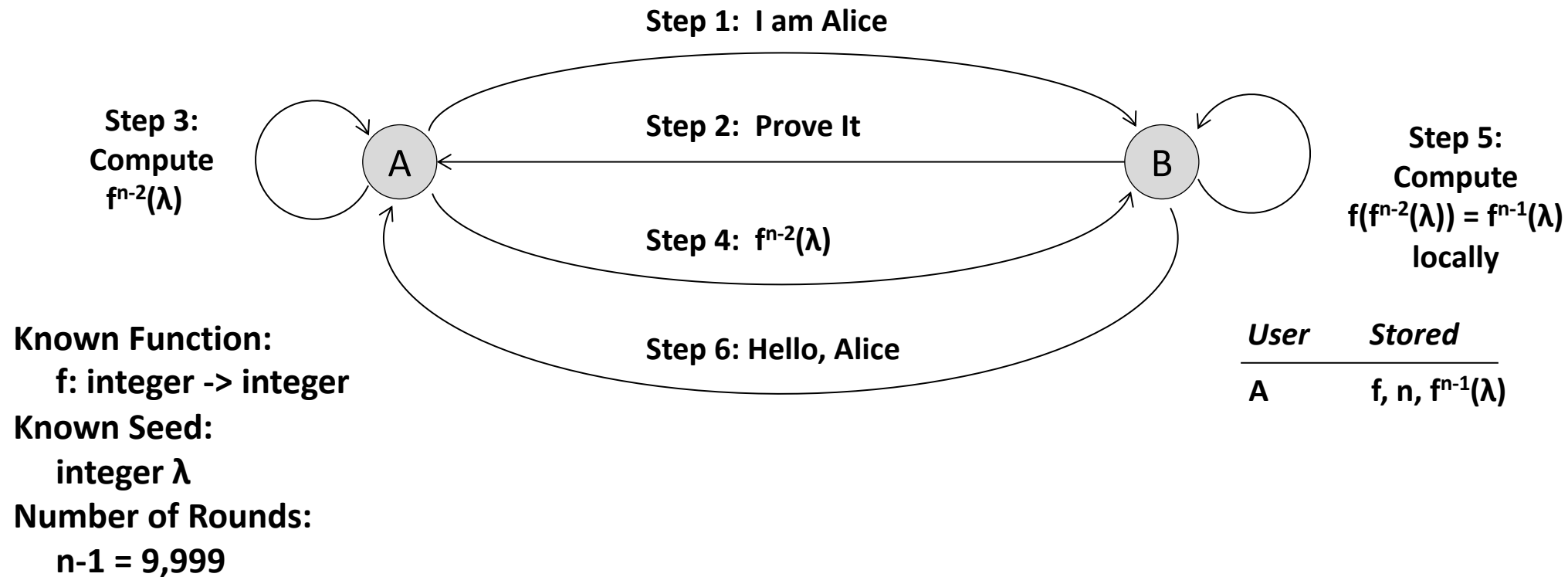
integer λ

Number of Rounds:

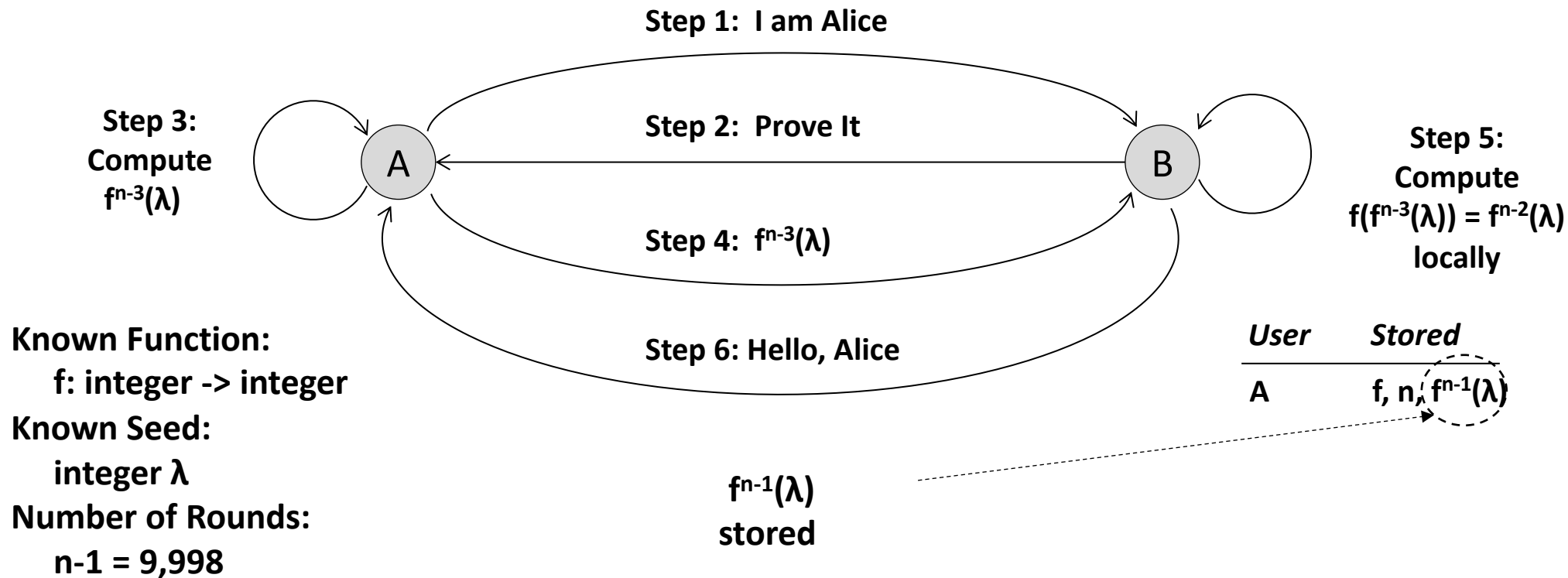
$n-1 = 9,999$

<i>User</i>	<i>Stored</i>
A	$f, n, f^{n-1}(\lambda)$

Lamport S/Key Protocol



Lamport S/Key Protocol



Lamport S/Key Protocol

A

B

Known Function:

$f: \text{integer} \rightarrow \text{integer}$

Known Seed:

integer λ

Number of Rounds:

$n-2 = 9,998$

$f^{n-2}(\lambda)$
now stored
(decremented)

User	Stored
A	$f, n, f^{n-2}(\lambda)$

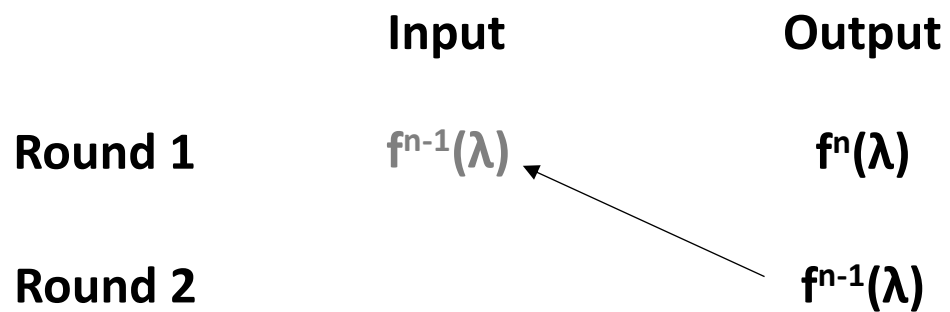
Lamport S/Key Protocol – Analysis

	Input	Output
Round 1	-	$f^n(\lambda)$

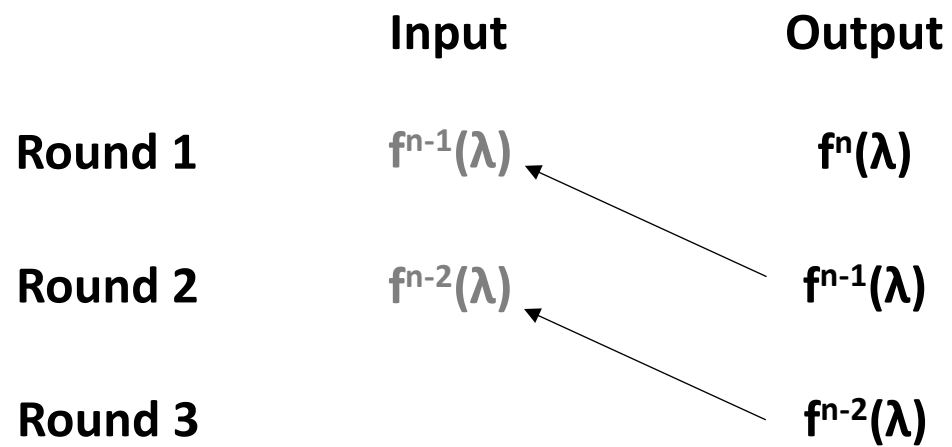
Lamport S/Key Protocol – Analysis

	Input	Output	
Round 1	-	$f^n(\lambda)$	
Round 2		$f^{n-1}(\lambda)$	Note: $f(f^{n-1}(\lambda)) = f^n(\lambda)$

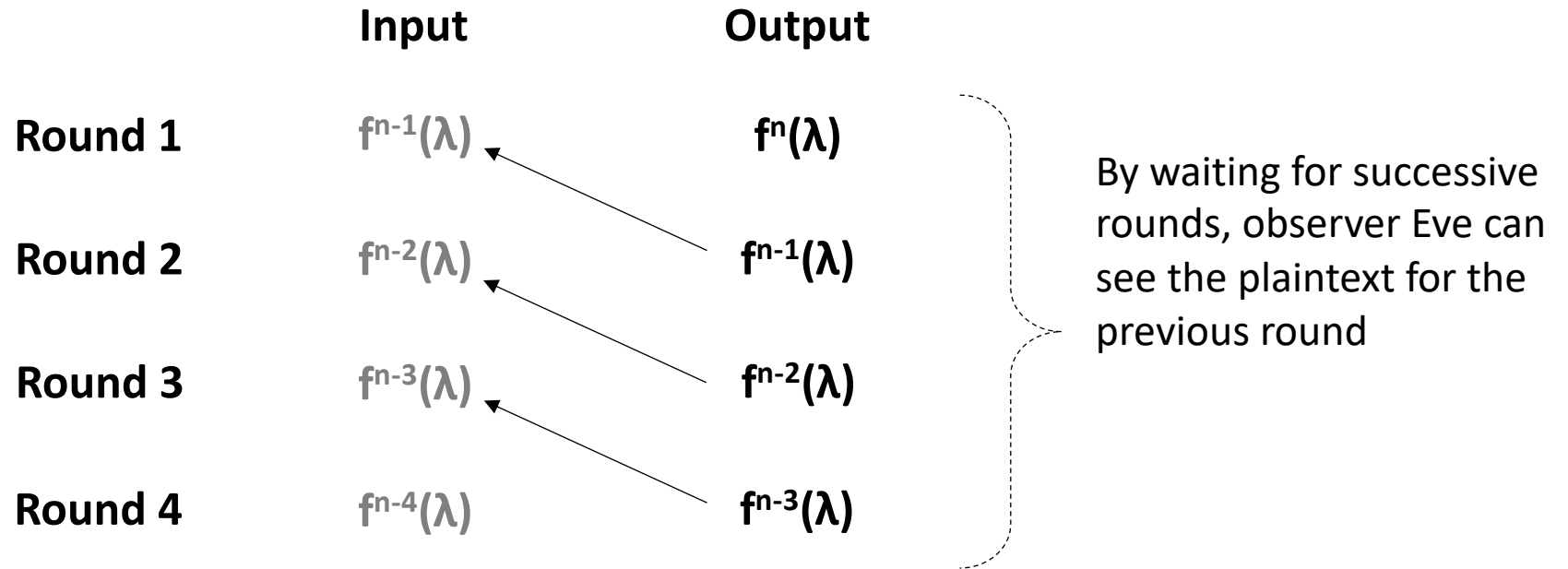
Lamport S/Key Protocol – Analysis



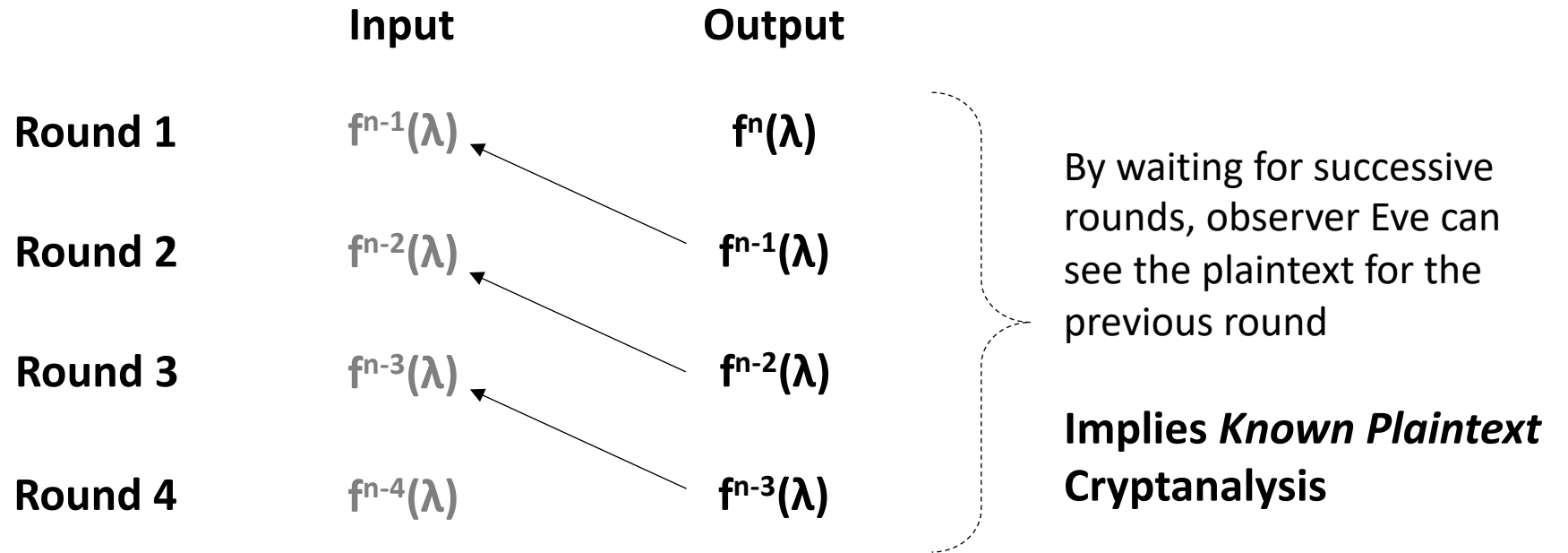
Lamport S/Key Protocol – Analysis



Lamport S/Key Protocol – Analysis



Lamport S/Key Protocol – Analysis



How Does Conventional Cryptography Work?

Definition: Cryptosystem

A cryptosystem is a five-tuple consisting of

- Encryption function E
- Decryption function D
- Set of plaintext elements P
- Set of ciphertext elements C
- Set of cryptographic keys K

$$E(p) = c$$

$$\{ p \} = c$$

$$D(c) = p$$

$$\{ c \} = p$$

$$D(E(p))$$

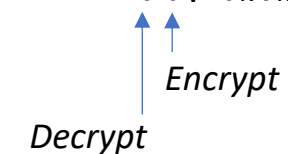
$$\{\{ p \}\} = p$$

$$E_k(p) = c$$

$$\{ p \}_k = c$$

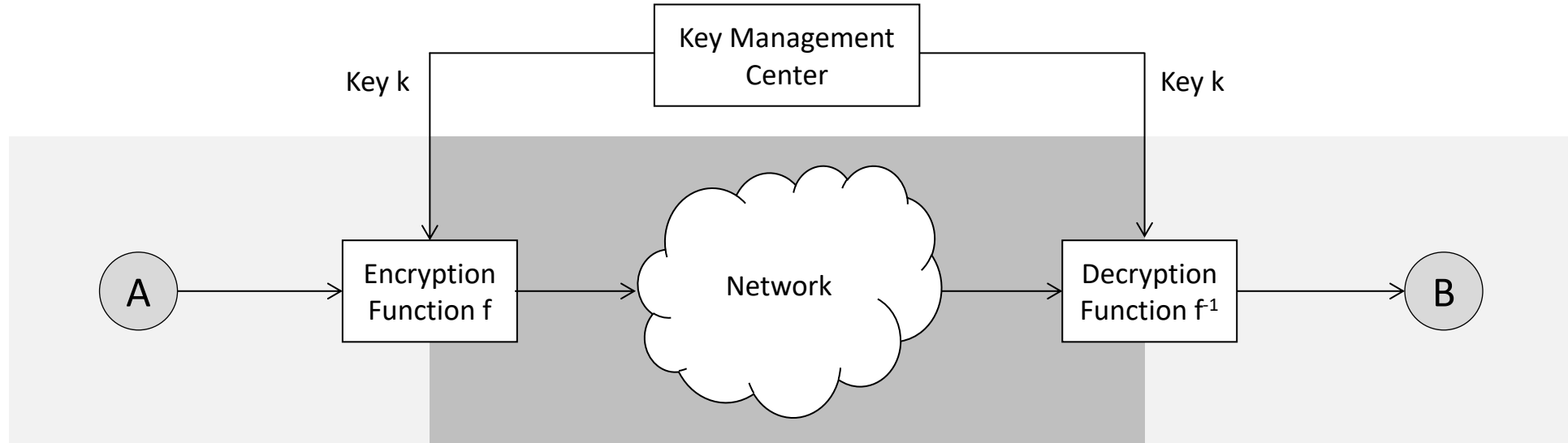
$$D_k(c) = p$$

$$\{\{ p \}_k\}_k = p$$

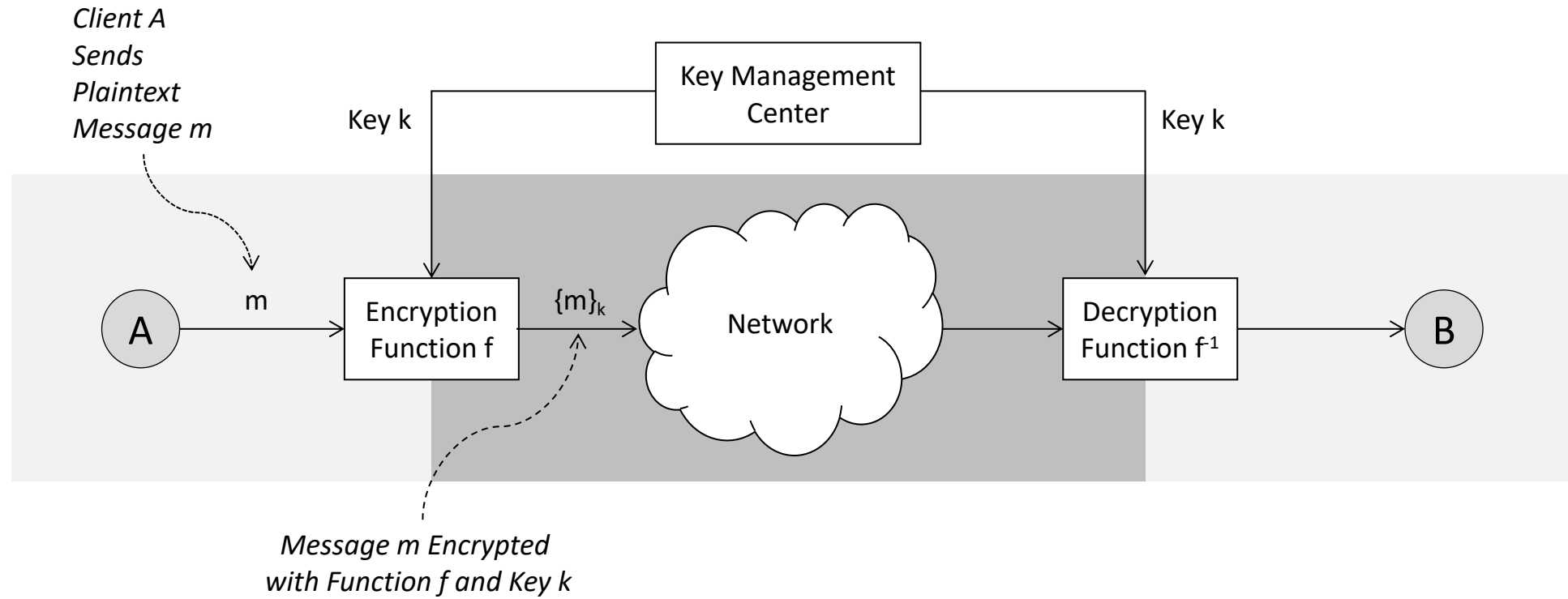


 Decrypt Encrypt

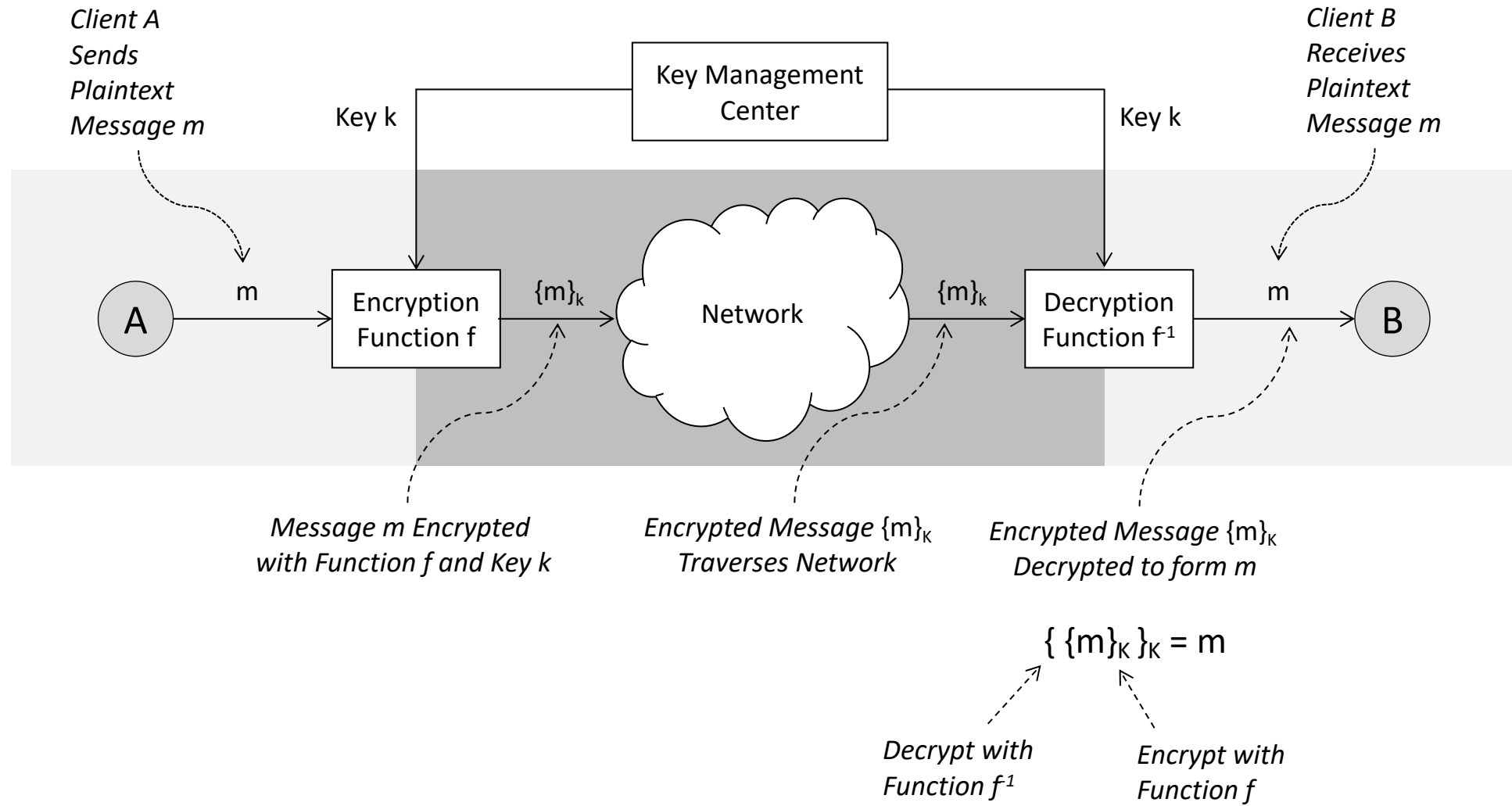
Conventional Encryption Schema



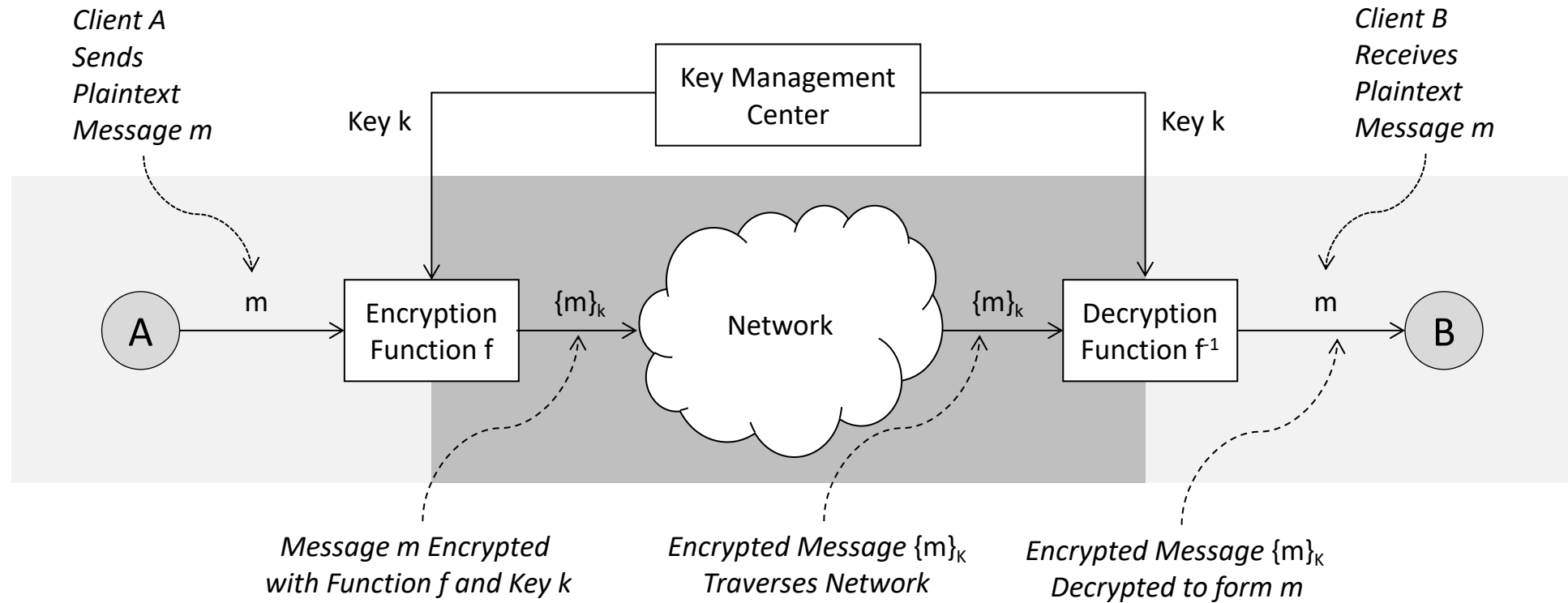
Conventional Encryption Schema



Conventional Encryption Schema



Conventional Encryption Schema



Two Important Security Properties:

1. Secrecy Between A and B
2. Authentication of A by B

$$\{ \{m\}_k \}_k = m$$

Decrypt with Function f^{-1} Encrypt with Function f

What is the Simplest Example Encryption Algorithm?

XOR Function

$$1 \text{ XOR } 1 = 0$$

$$0 \text{ XOR } 0 = 0$$

$$1 \text{ XOR } 0 = 1$$

$$0 \text{ XOR } 1 = 1$$

Conventional Encryption Algorithm – Simplest Example

XOR Function

Plaintext Input 0 0 1 0 0 0 1 1

1 XOR 1 = 0

0 XOR 0 = 0

1 XOR 0 = 1

0 XOR 1 = 1

Conventional Encryption Algorithm – Simplest Example

XOR Function

Plaintext Input 0 0 1 0 0 0 1 1

Key 1 1 1 0 1 1 1 0

1 XOR 1 = 0

0 XOR 0 = 0

1 XOR 0 = 1

0 XOR 1 = 1

Conventional Encryption Algorithm – Simplest Example

XOR Function

Plaintext Input 0 0 1 0 0 0 1 1

Key 1 1 1 0 1 1 1 0

Ciphertext 1 1 0 0 1 1 0 1

1 XOR 1 = 0

0 XOR 0 = 0

1 XOR 0 = 1

0 XOR 1 = 1

Conventional Encryption Algorithm – Simplest Example

XOR Function

Plaintext Input 0 0 1 0 0 0 1 1

Key 1 1 1 0 1 1 1 0

Ciphertext 1 1 0 0 1 1 0 1

Key 1 1 1 0 1 1 1 0

1 XOR 1 = 0

0 XOR 0 = 0

1 XOR 0 = 1

0 XOR 1 = 1

Conventional Encryption Algorithm – Simplest Example

XOR Function

Plaintext Input 0 0 1 0 0 0 1 1

Key 1 1 1 0 1 1 1 0

Ciphertext 1 1 0 0 1 1 0 1

Key 1 1 1 0 1 1 1 0

Plaintext Output 0 0 1 0 0 0 1 1

1 XOR 1 = 0

0 XOR 0 = 0

1 XOR 0 = 1

0 XOR 1 = 1

Conventional Encryption Algorithm – Simplest Example

What are the Two Most Basic Design Strategies for Encryption Algorithms?

Substitution Cipher – Replacement of one or more things with one or more things (Symmetric versus Asymmetric)

Source: Thomas Carlyle

MEN'S	HEARTS	OUGHTY	NOT	TO	BE	SET
ONB	H	MNPTZH	UIJMZ	BUZ	ZU	XN
1	10	18	4	5	10	9
12	8	8	2	3	5	12

AGAINST	ONE	ANOTHER,	BUT	SET	WITH	ONE
PJPKBHZ	UBN	PBUZMNT	XIZ	HNZ	VKZM	UBN
9	3	9	4	10	8	12
8	12	10	8	10	8	12

ANOTHER,	AND	ALL	AGAINST	EVIL	ONLY,
PBUZMNT	PBW	PFF	PJPKBHZ	NAKF	URFD
9	10	8	12	9	10
3	3	10	1	9	4

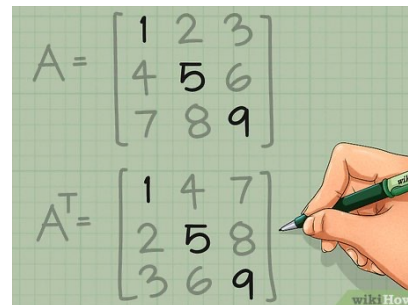
State	
a	1
b	10
c	1
d	1
e	4
f	6
g	2
h	3
i	3
j	4
k	5
l	10
m	1
n	9
o	2
p	6
q	1
r	1
s	1
t	12

Conventional Encryption Algorithm – Strategies

Substitution Cipher – Replacement of one or more things with one or more things (Symmetric versus Asymmetric)



Transposition Cipher – Use of matrix arithmetic to represent and manipulate text (Linear Algebraic basis)



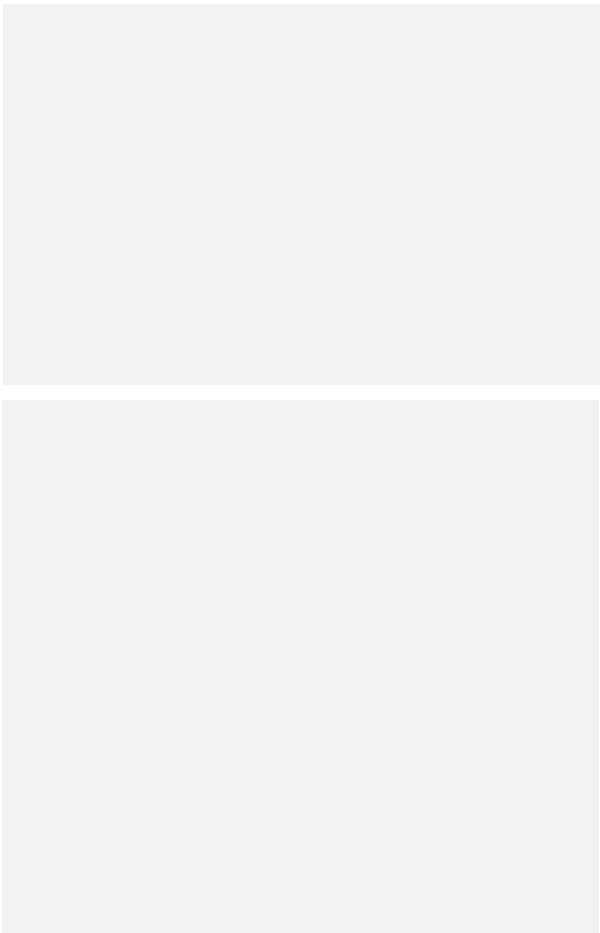
Conventional Encryption Algorithm – Strategies

What is the Data Encryption Standard (DES)?
(How Did It Influence AES?)

Data Encryption Standard (DES)

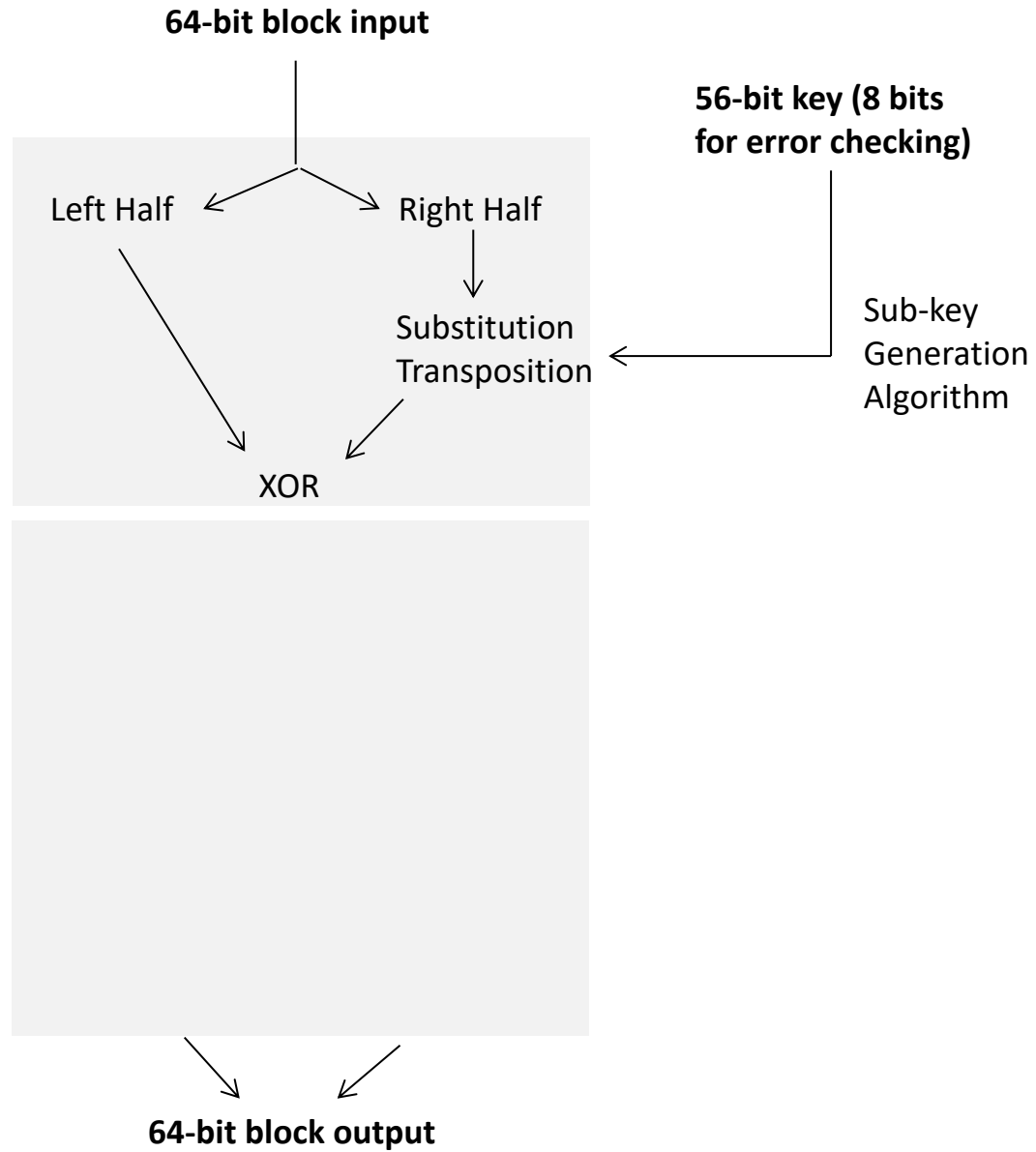
64-bit block input

56-bit key (8 bits
for error checking)

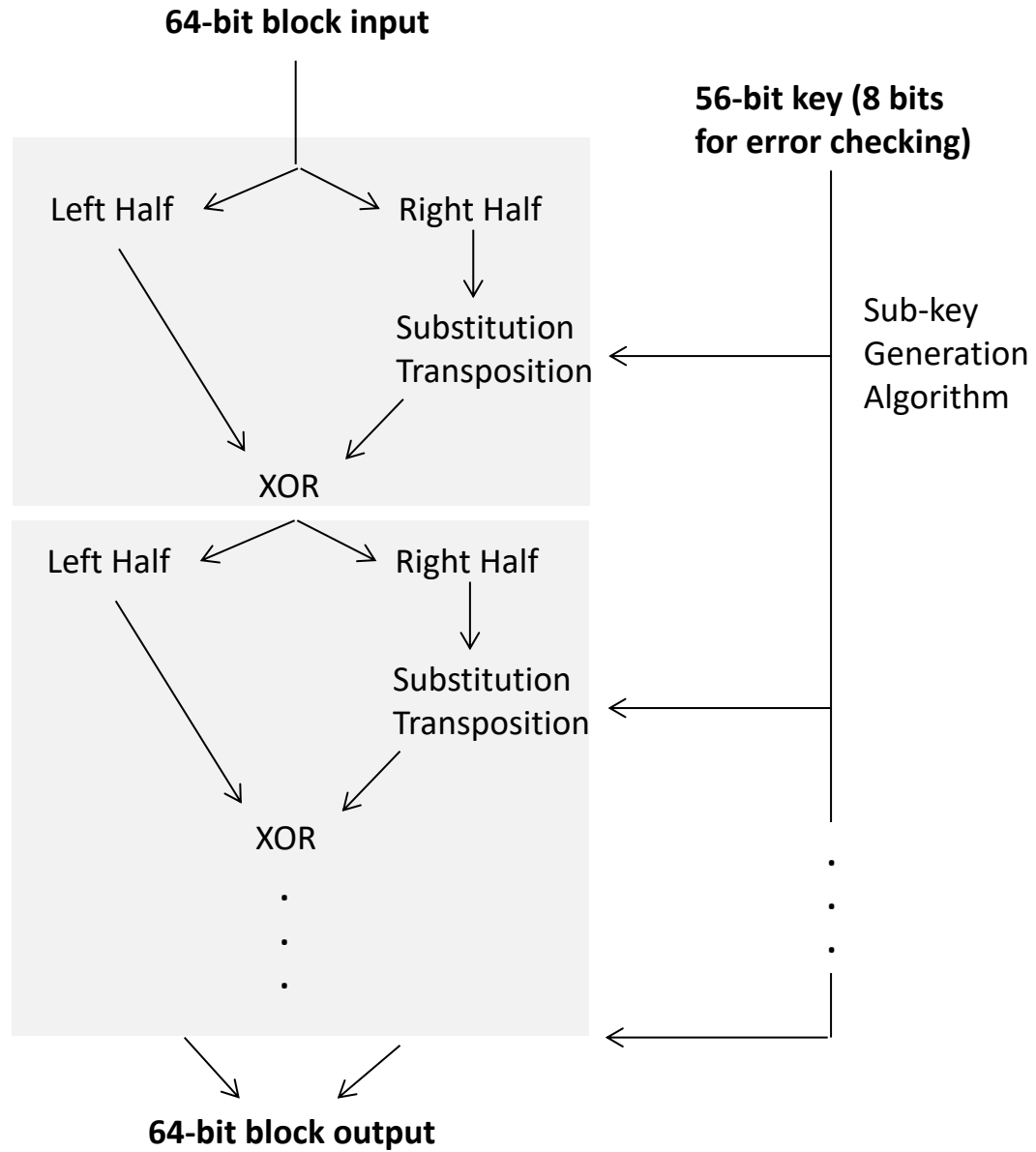


64-bit block output

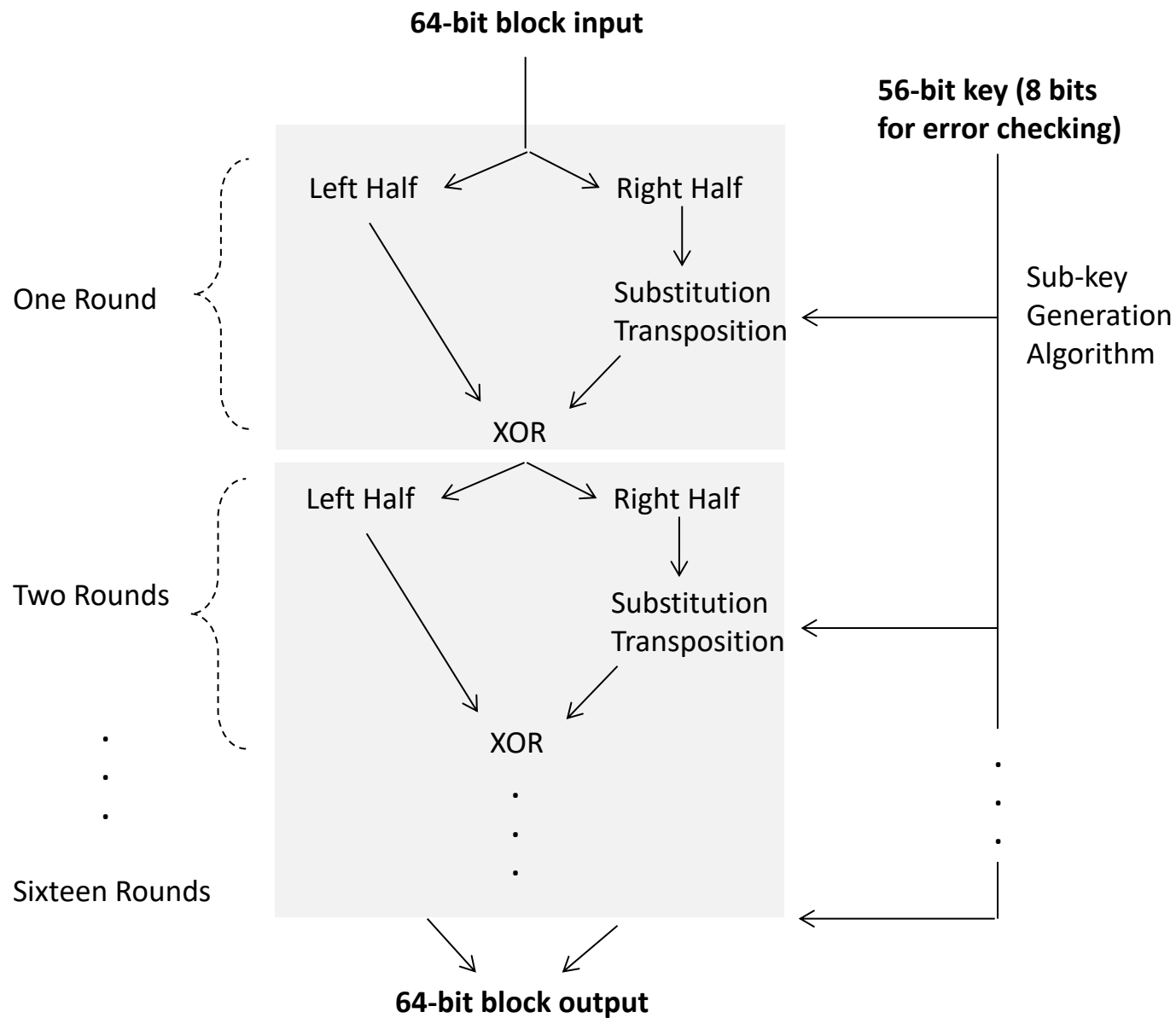
Data Encryption Standard (DES)



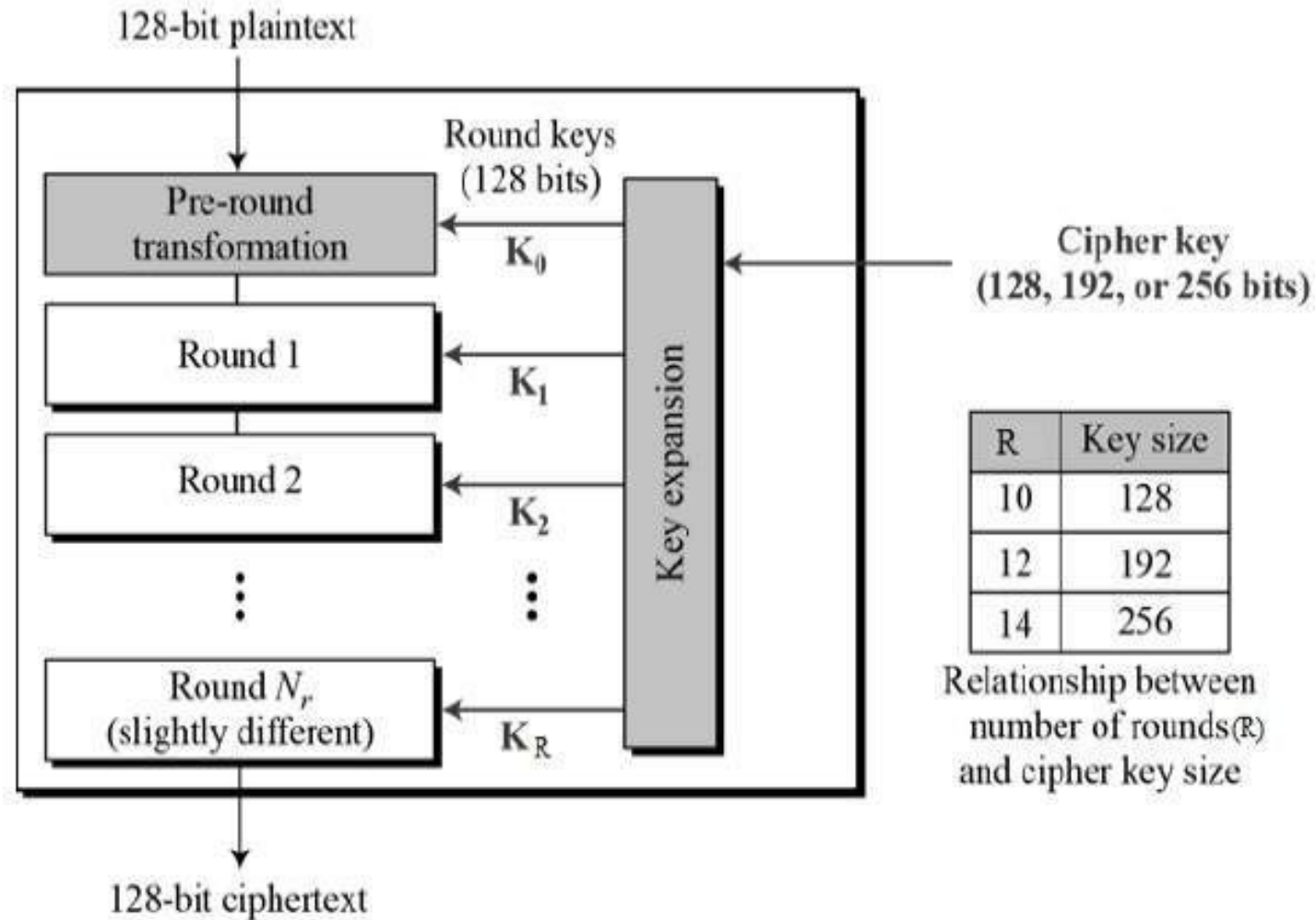
Data Encryption Standard (DES)



Data Encryption Standard (DES)



Advanced Encryption Standard (AES)



What is Triple DES (3DES)?
(How Did It Solve Key Length Issues and
1DES Interoperability?)

Triple-DES

$\{ m \}_{K1}$	Single-DES	56 Bit Key
----------------	------------	------------

Triple-DES

$\{ m \}_{K1}$	Single-DES	56 Bit Key
----------------	------------	------------

$\{ \{ m \}_{K1} \}_{K2}$ Double-DES 112 Bit Key

Triple-DES

 $\{ m \}_{K_1}$

Single-DES

56 Bit Key

 $\{ \{ m \}_{K_1} \}_{K_2}$

Double-DES

112 Bit Key

 $\{ \{ \{ m \}_{K_1} \}_{K_2} \}_{K_3}$

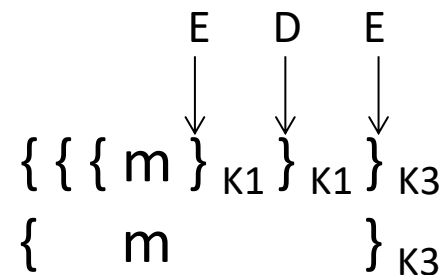
Triple-DES

168 Bit Key

Triple-DES

$\{ m \}_{K1}$	Single-DES	56 Bit Key
$\{ \{ m \}_{K1} \}_{K2}$	Double-DES	112 Bit Key
$\{ \{ \{ m \}_{K1} \}_{K2} \}_{K3}$	Triple-DES	168 Bit Key

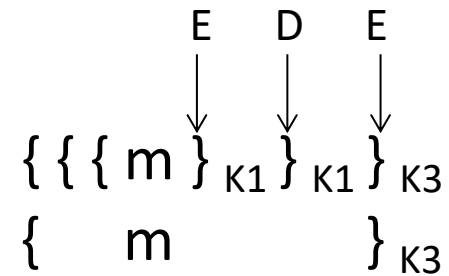
Single-DES Mode: $K1 = K2 \neq K3$



Triple-DES

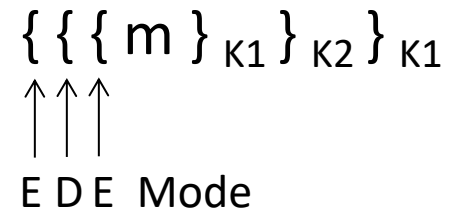
$\{ m \}_{K_1}$	Single-DES	56 Bit Key
$\{ \{ m \}_{K_1} \}_{K_2}$	Double-DES	112 Bit Key
$\{ \{ \{ m \}_{K_1} \}_{K_2} \}_{K_3}$	Triple-DES	168 Bit Key

Single-DES Mode: $K_1 = K_2 \neq K_3$



Triple-DES Mode: $K_1 = K_3 \neq K_2$

Effective 112 bits

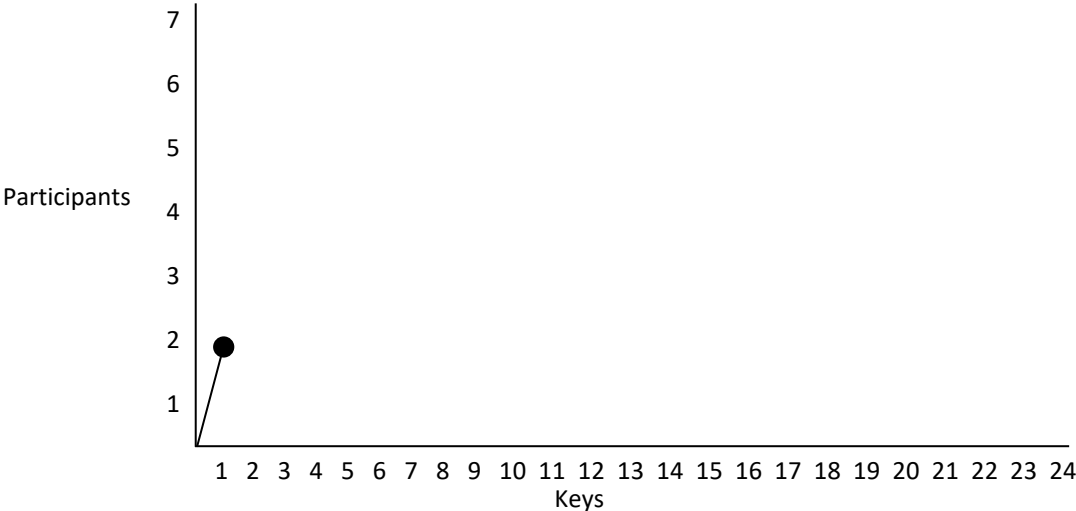


What is the Scaling Issue for
Conventional Cryptography?

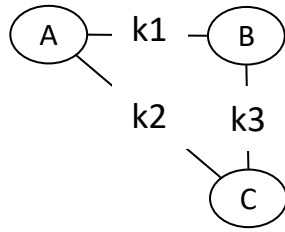
Conventional Encryption Scaling Issue



2 participants – 1 shared key



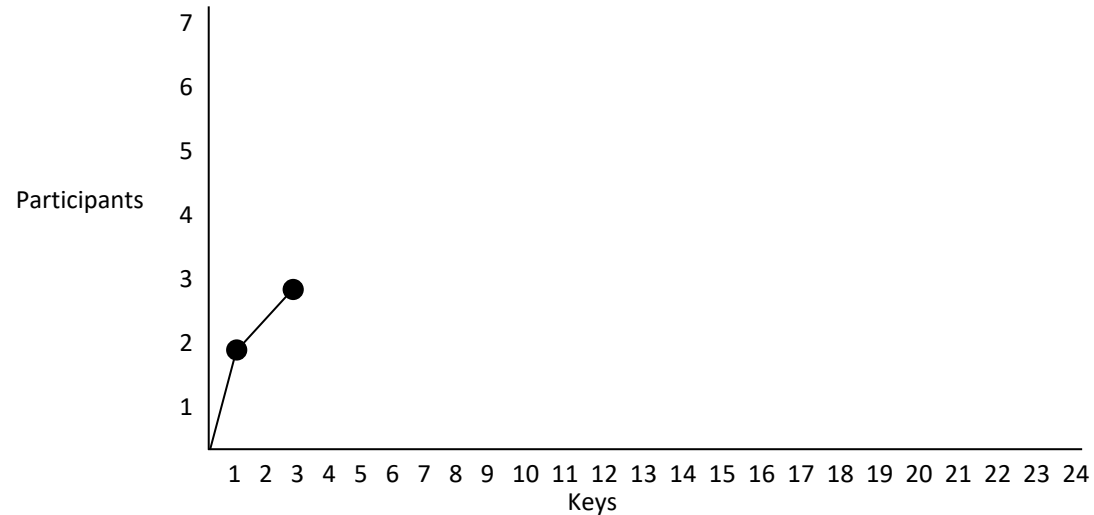
Conventional Encryption Scaling Issue



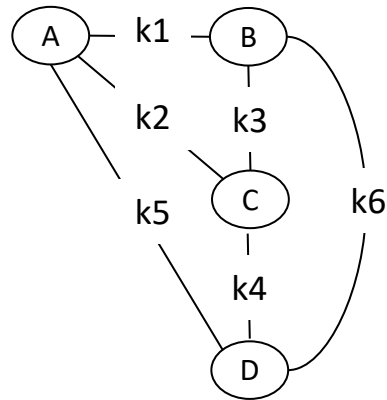
2 participants – 1 shared key

3 participants – 3 shared keys

Added participant 1
Added new keys 2



Conventional Encryption Scaling Issue

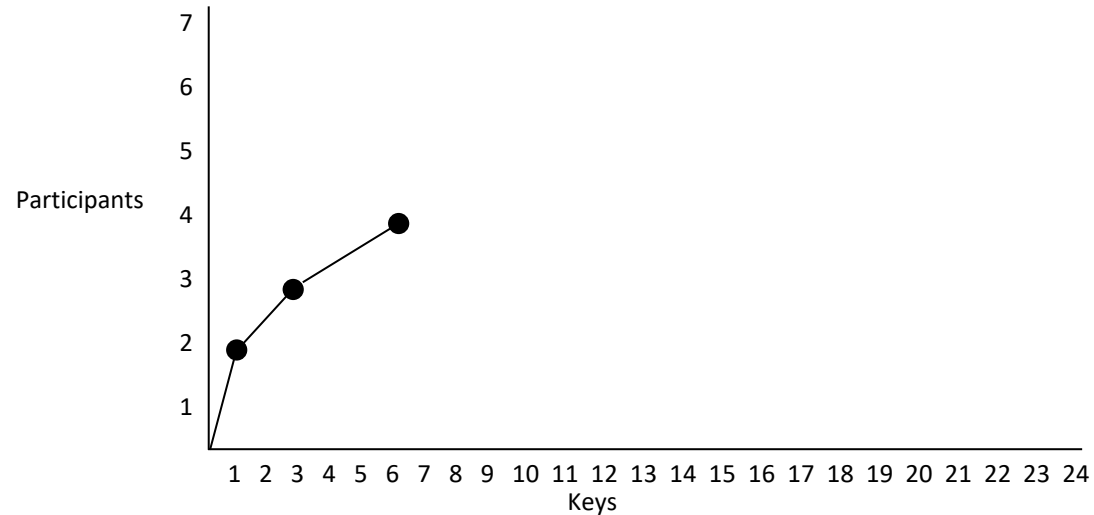


2 participants – 1 shared key

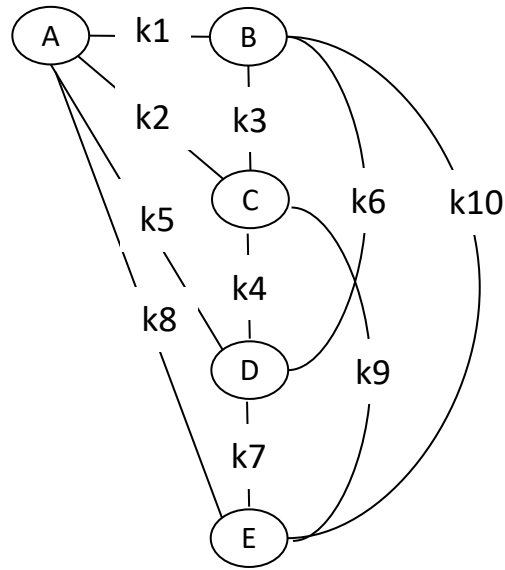
3 participants – 3 shared keys

4 participants – 6 shared keys

Added participant 1
Added new keys 3

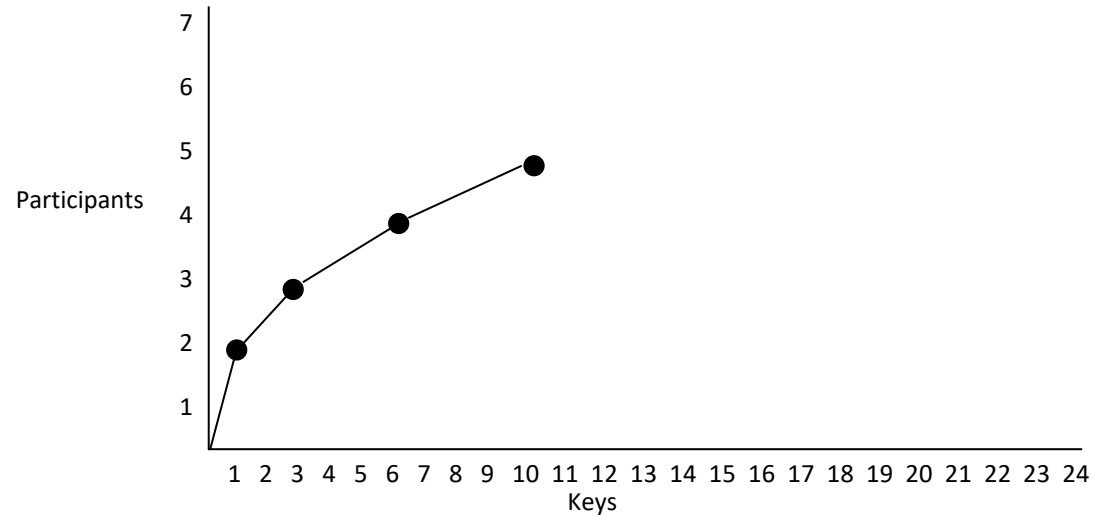


Conventional Encryption Scaling Issue

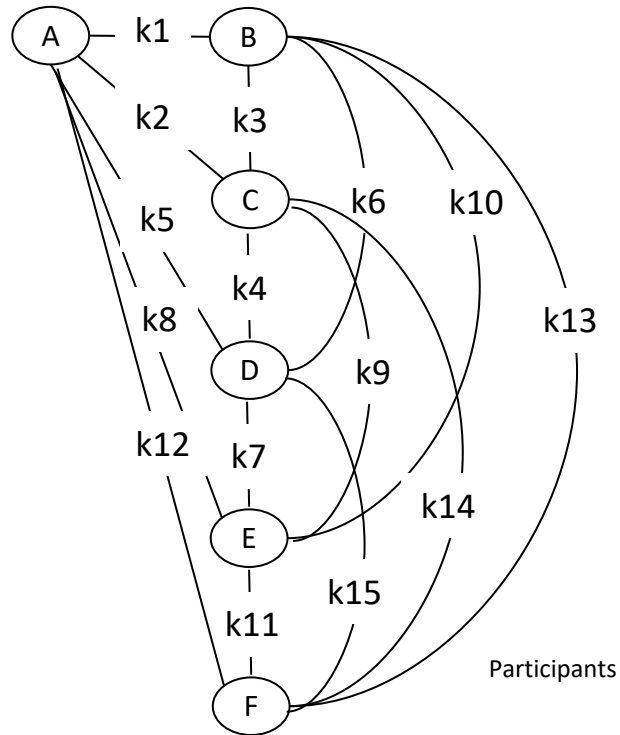


- 2 participants – 1 shared key
- 3 participants – 3 shared keys
- 4 participants – 6 shared keys
- 5 participants – 10 shared keys

Added participant 1
Added new keys 4

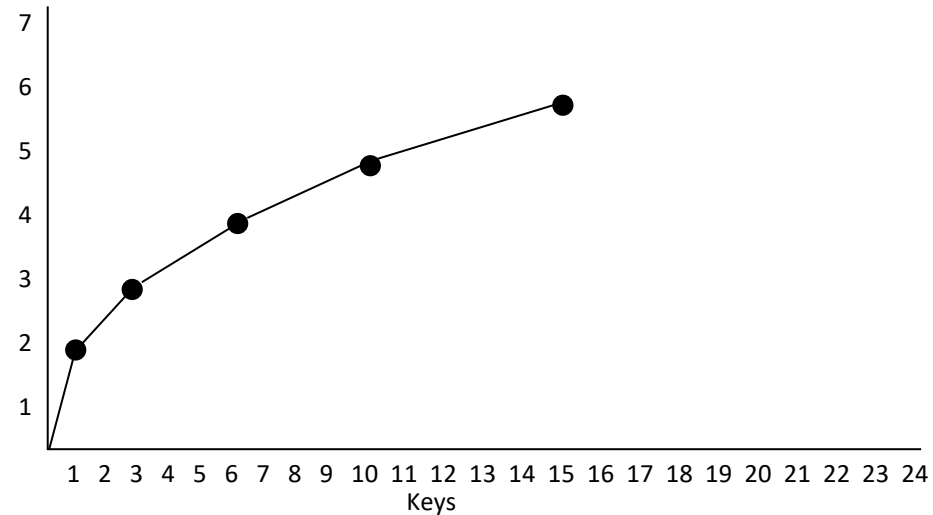


Conventional Encryption Scaling Issue

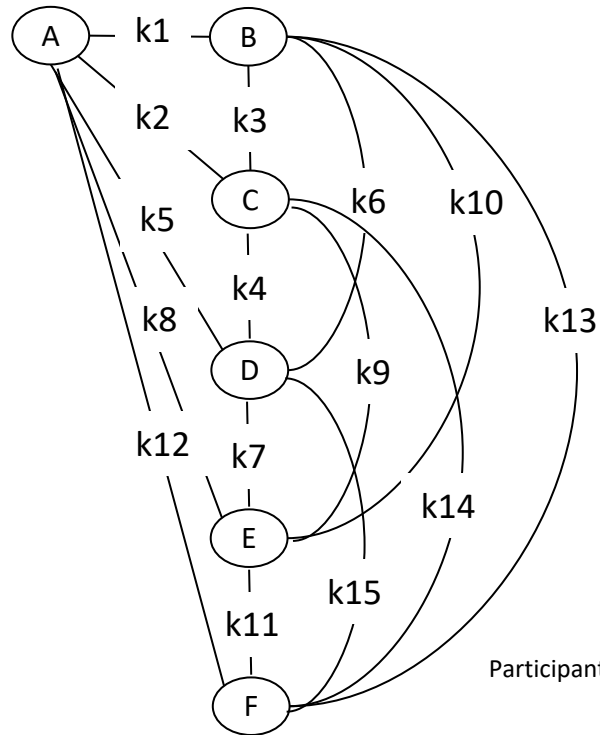


Added participant 1
Added new keys 5

2 participants – 1 shared key
3 participants – 3 shared keys
4 participants – 6 shared keys
5 participants – 10 shared keys
6 participants – 15 shared keys

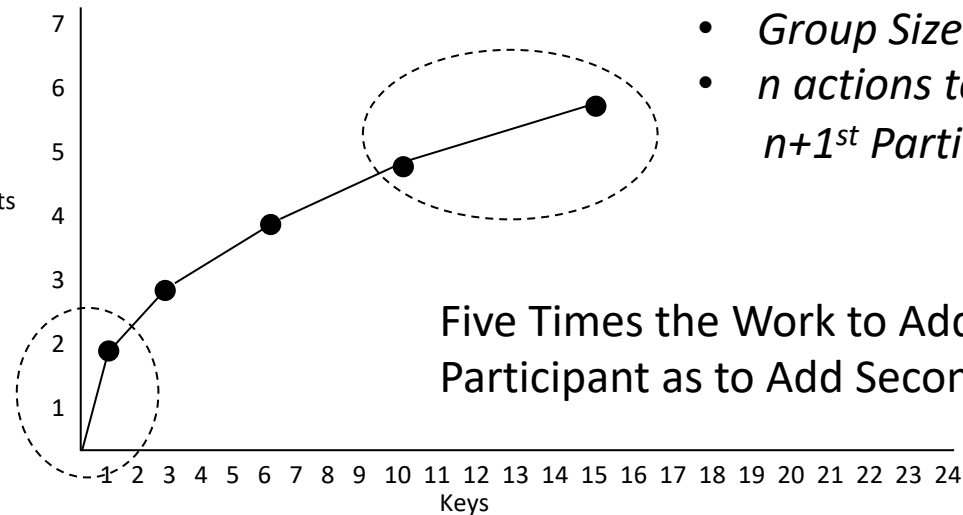


Conventional Encryption Scaling Issue



Added participant 1
Added new keys 5

2 participants – 1 shared key
3 participants – 3 shared keys
4 participants – 6 shared keys
5 participants – 10 shared keys
6 participants – 15 shared keys

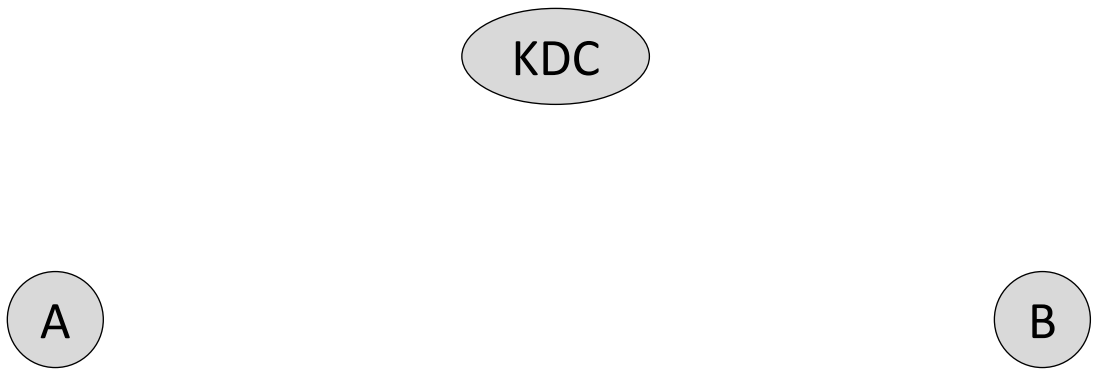


- *Group Size = n*
- *n actions to add $n+1^{st}$ Participant*

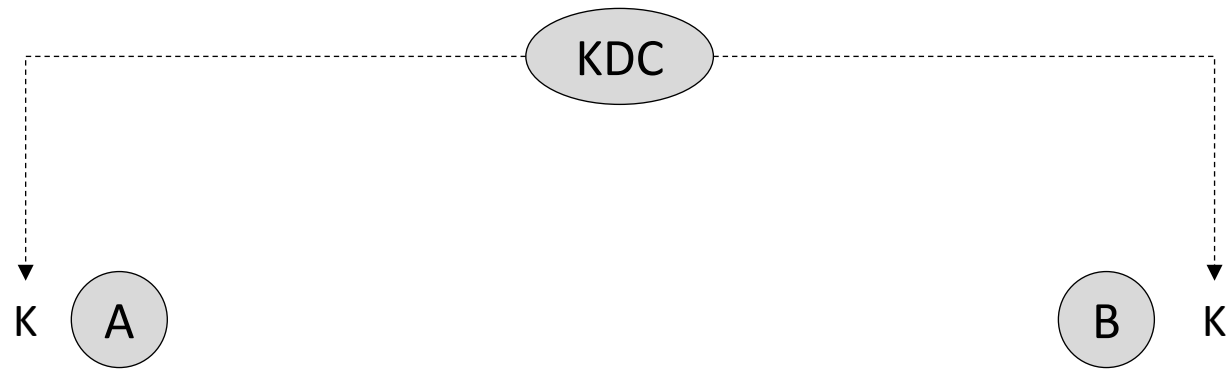
Five Times the Work to Add Sixth Participant as to Add Second

What are the Key Security Properties of
Conventional Cryptography?

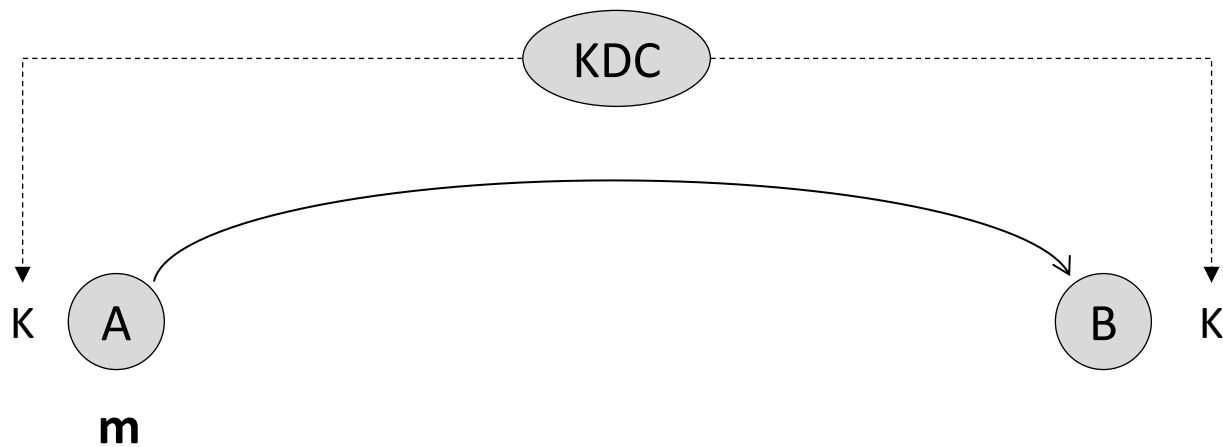
Conventional Cryptography



Conventional Cryptography

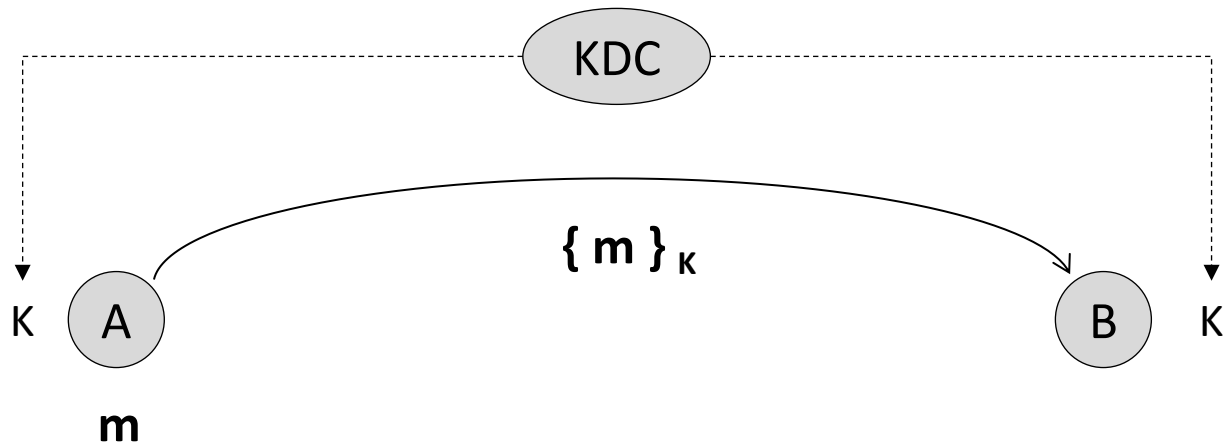


Conventional Cryptography



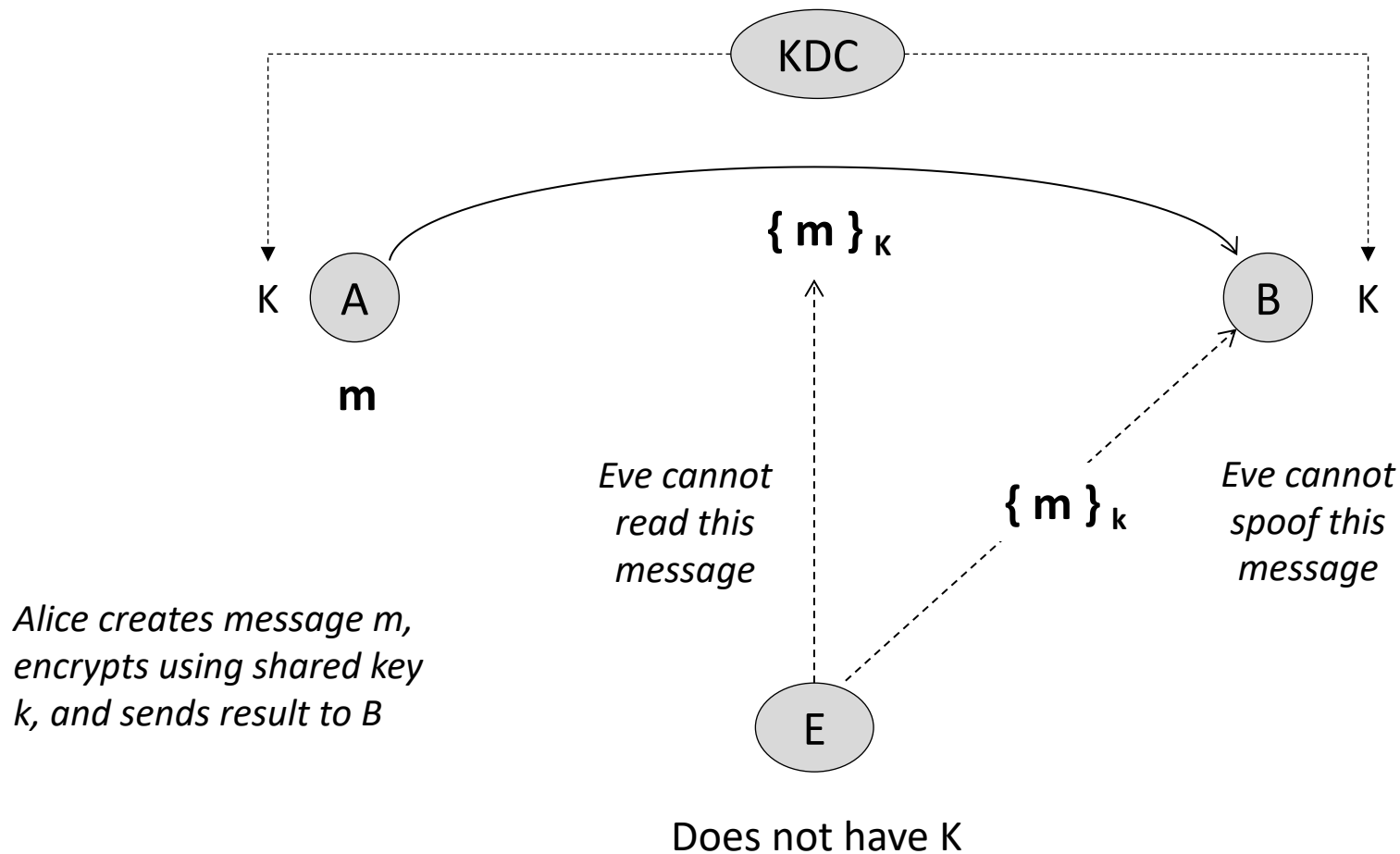
Alice creates message m . . .

Conventional Cryptography

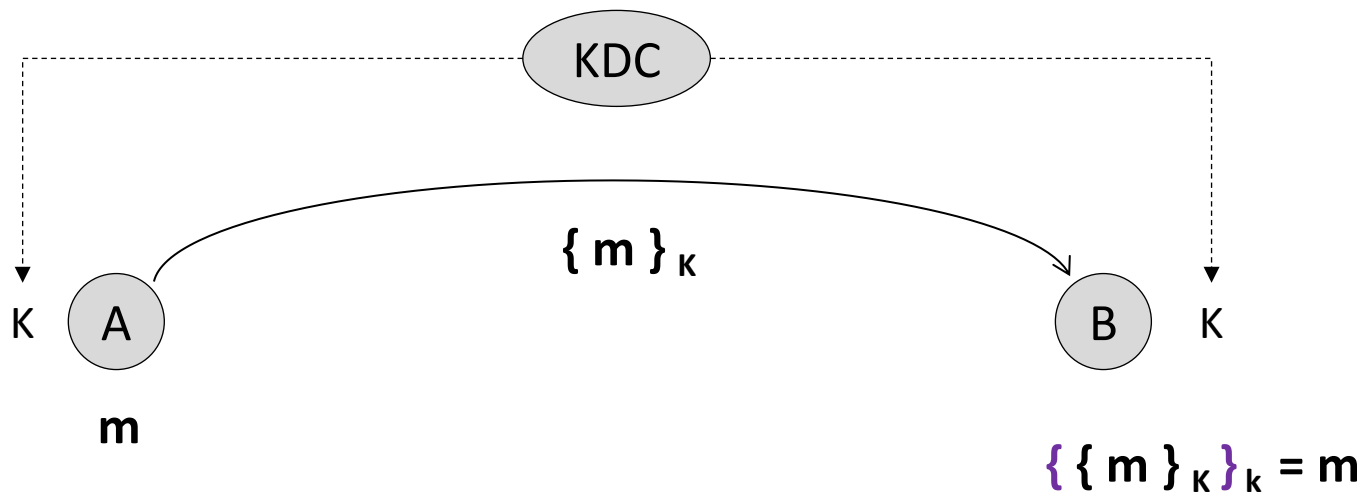


*Alice creates message m ,
encrypts using shared key
 k , and sends result to B*

Conventional Cryptography

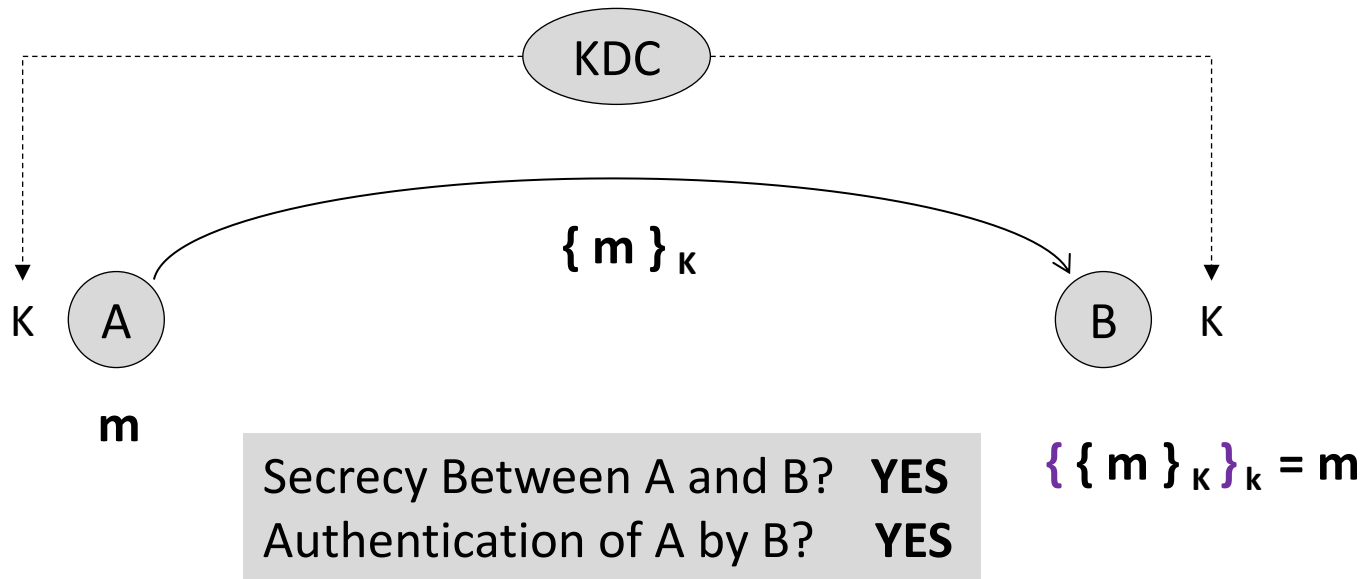


Conventional Cryptography

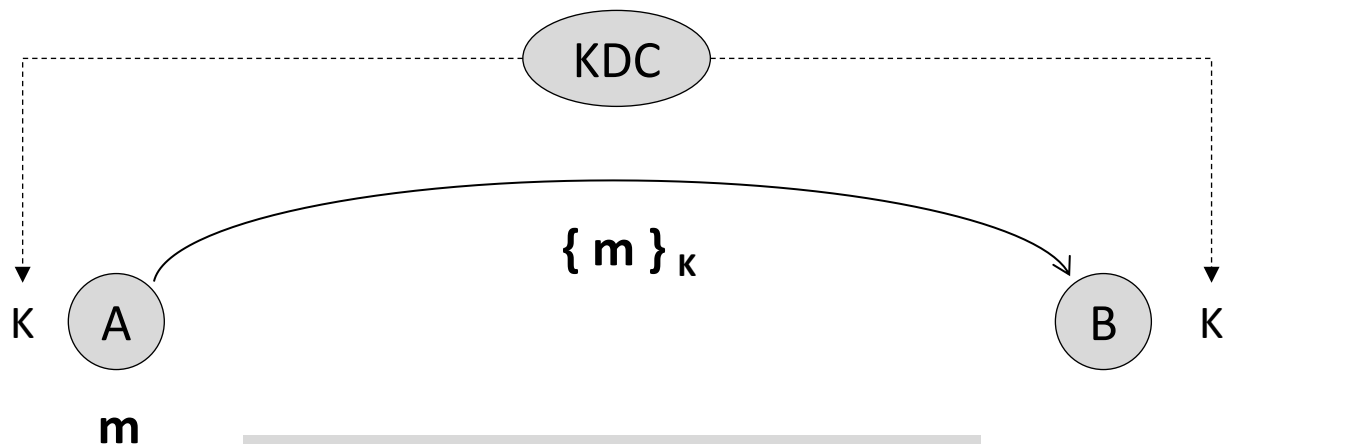


Bob receives encrypted message, and decrypts using shared key k , and obtains message m

Conventional Cryptography



Conventional Cryptography



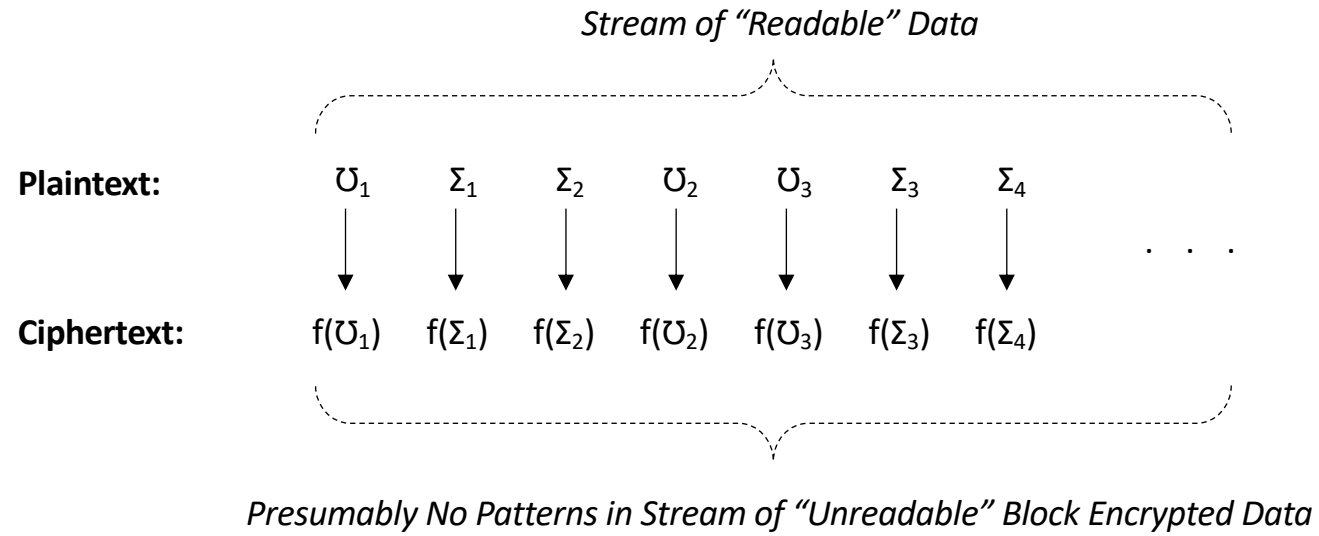
Secrecy Between A and B? **YES**
 Authentication of A by B? **YES**

$$\{\{m\}_K\}_K = m$$

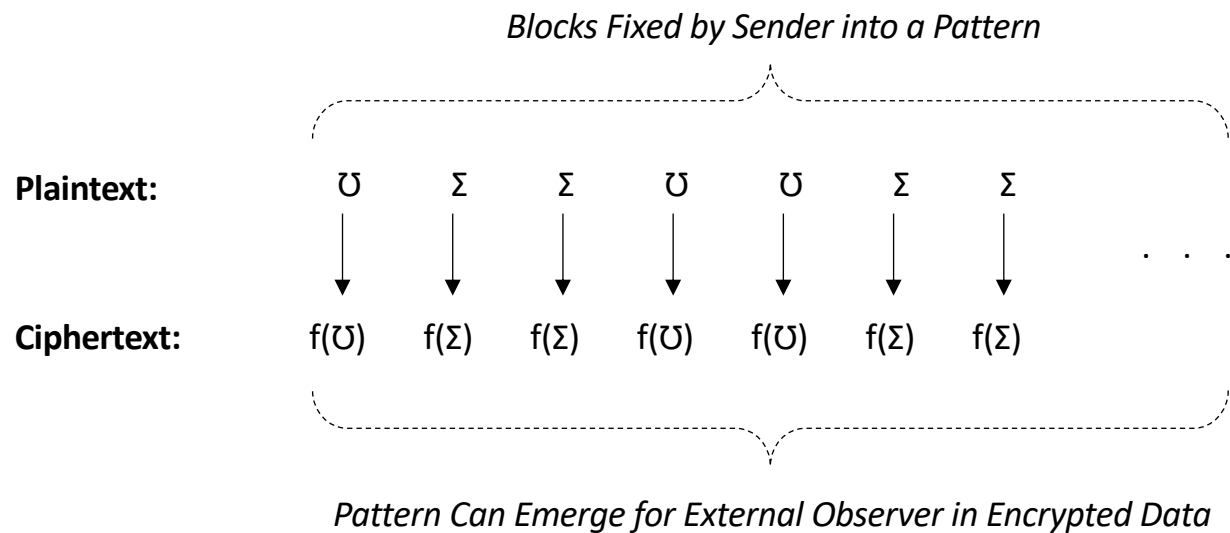
Does this approach scale? **NO**

How Does Block Chaining Work?

Conventional Block Cryptography



Conventional Block Cryptography – Covert Channel



Conventional Block Cryptography – 1 bps Channel

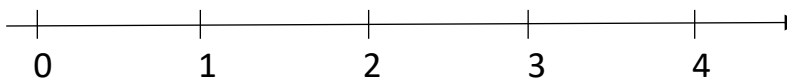
Plaintext:

0 1 1 0 0 . . .

↓ ↓ ↓ ↓ ↓

Ciphertext:

$f(0) = x$ $f(1) = y$ $f(1) = y$ $f(0) = x$ $f(0) = x$



Seconds

Block Chain Mode Cryptography – Circa 1976 at IBM

Patents

[Find prior art](#)[Discuss this patent](#)

Message verification and transmission error detection by block chaining

US 4074066 A

ABSTRACT

A message transmission system for the secure transmission of multi-block data messages from a sending station to a receiving station.

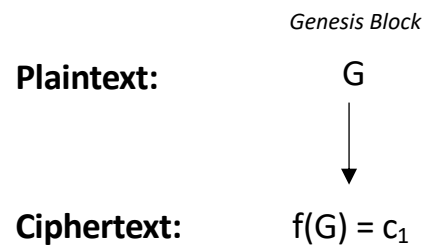
The sending station contains cryptographic apparatus operative in successive cycles of operation during each of which an input block of clear data bits is ciphered under control of an input set of cipher key bits to generate an output block of ciphered data bits for transmission to the receiving station. Included in the cryptographic apparatus of the sending station is means providing one of the inputs for each succeeding ciphering cycle of operation as a function of each preceding ciphering cycle of operation. As a result, each succeeding output block of ciphered data bits is effectively chained to all preceding cycles of operation of the cryptographic apparatus of the sending station and is a function of the corresponding input block of clear data bits, all preceding input blocks of clear data bits and the initial input set of cipher key bits.

Publication number	US4074066 A
Publication type	Grant
Application number	US 05/680,404
Publication date	Feb 14, 1978
Filing date	Apr 26, 1976
Priority date [?]	Apr 26, 1976
Also published as	CA1100588A, CA1100588A1, DE2715631A1, DE2715631C2
Inventors	William F. Ehrtam, Carl H. W. Meyer, John L. Smith, Walter L. Tuchman
Original Assignee	International Business Machines Corporation
Export Citation	BiBTeX, EndNote, RefMan
Patent Citations (5), Referenced by (52), Classifications (10)	
External Links: USPTO , USPTO Assignment , Espacenet	

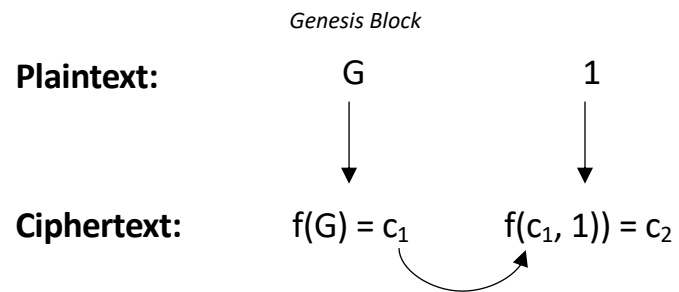
IMAGES (5)



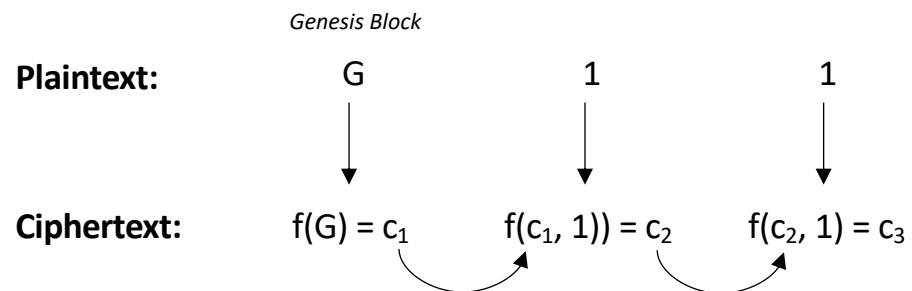
Block Chain Mode Cryptography



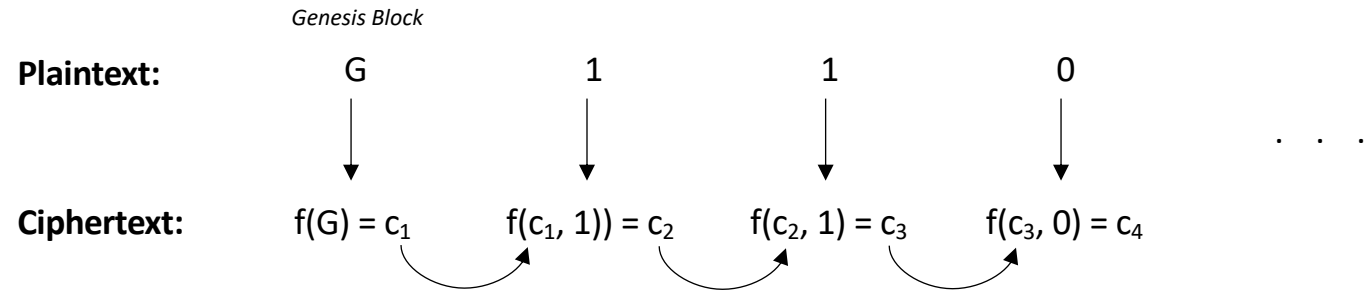
Block Chain Mode Cryptography



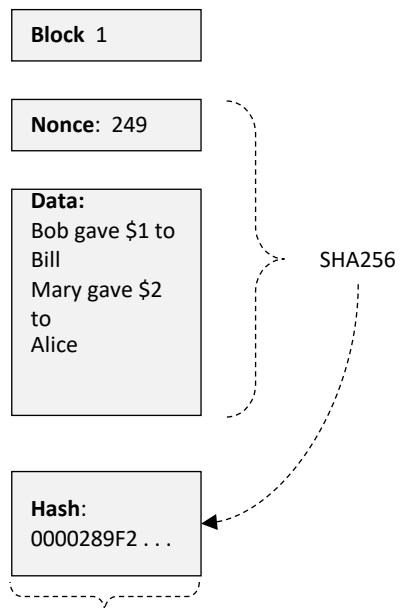
Block Chain Mode Cryptography



Block Chain Mode Cryptography

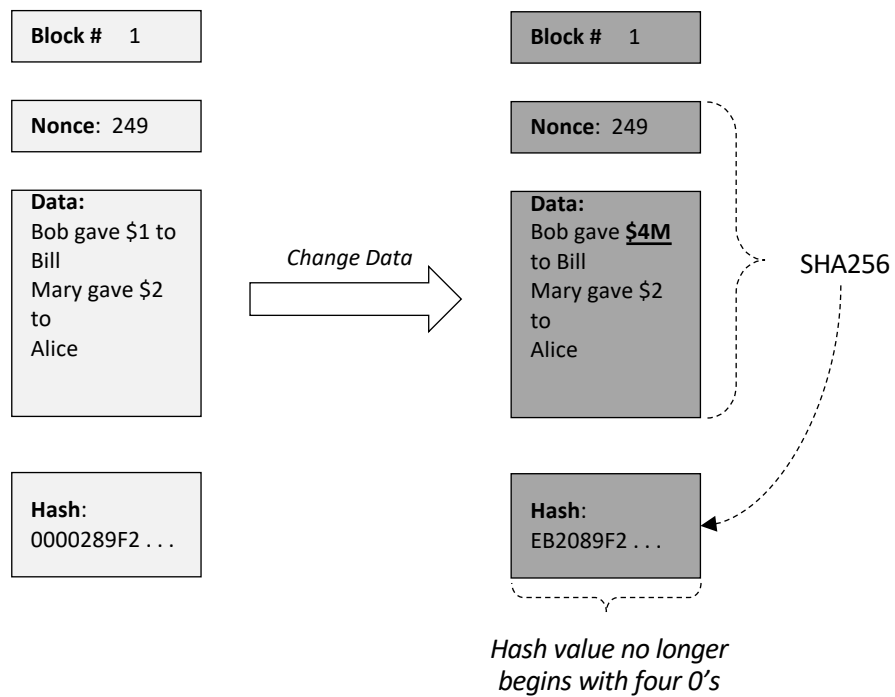


Modern Block Chain Usage

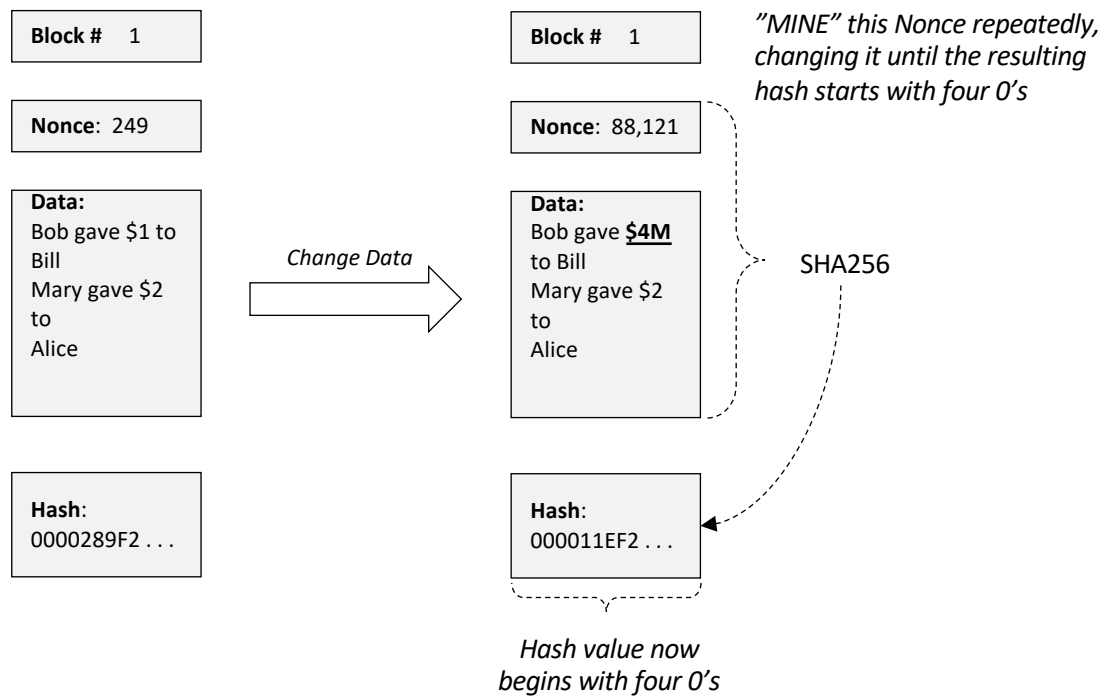


Hash value happens to begin with four 0's

Modern Block Chain Usage



Modern Block Chain Usage



Modern Block Chain Usage

Block # 1

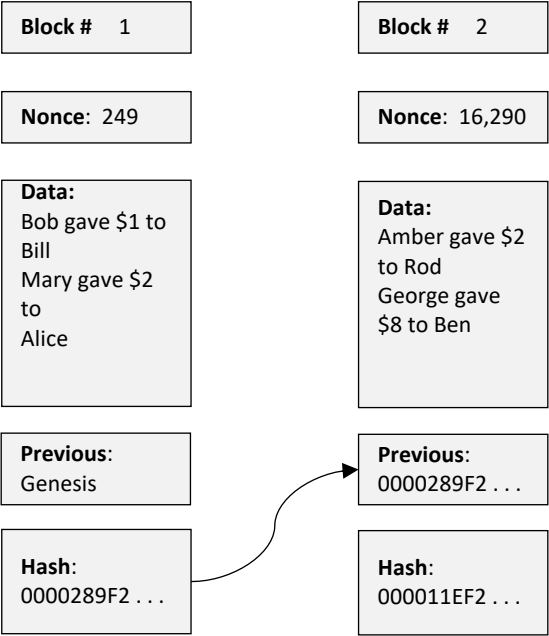
Nonce: 249

Data:
Bob gave \$1 to
Bill
Mary gave \$2
to
Alice

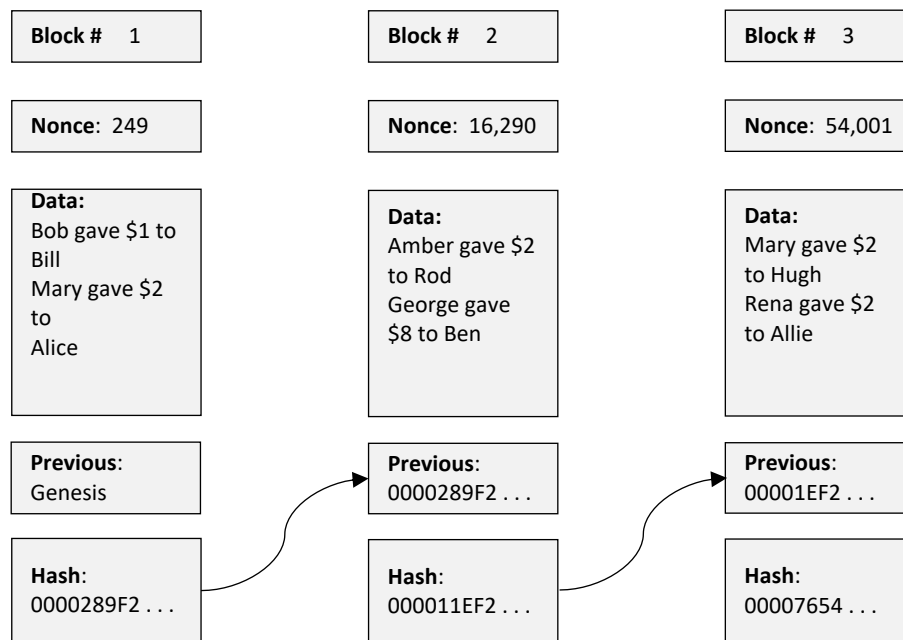
Previous:
Genesis

Hash:
0000289F2 . . .

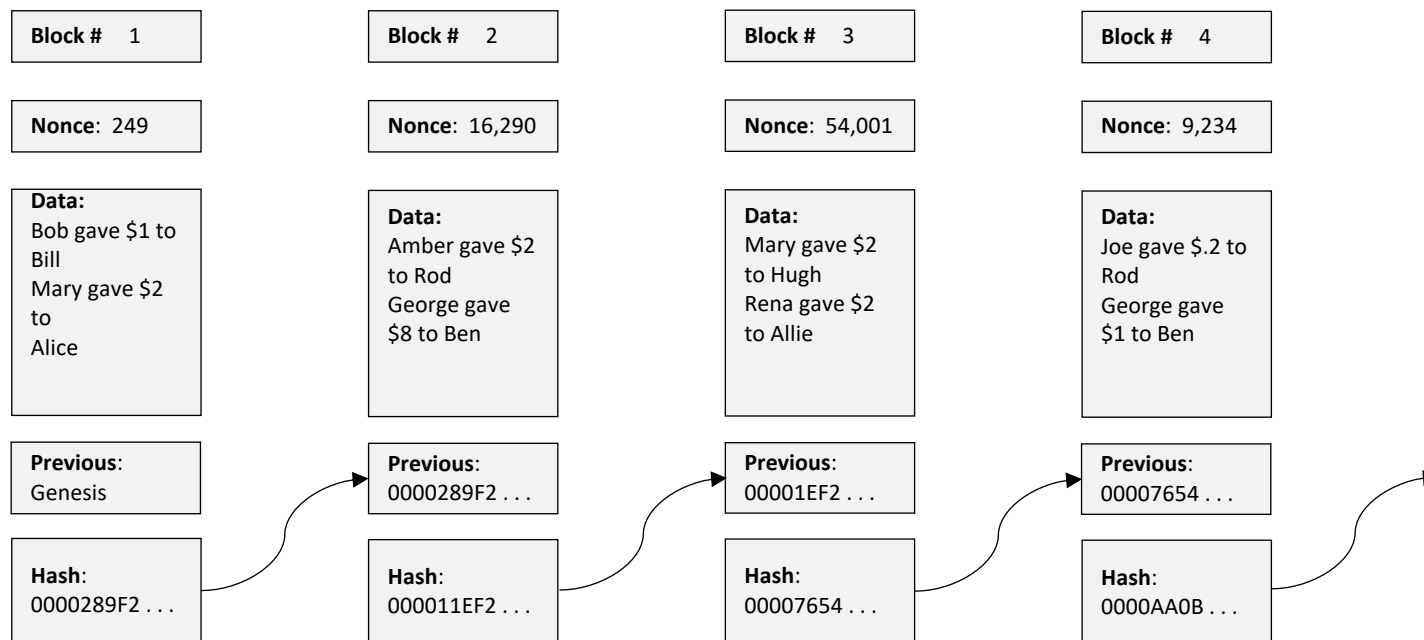
Modern Block Chain Usage



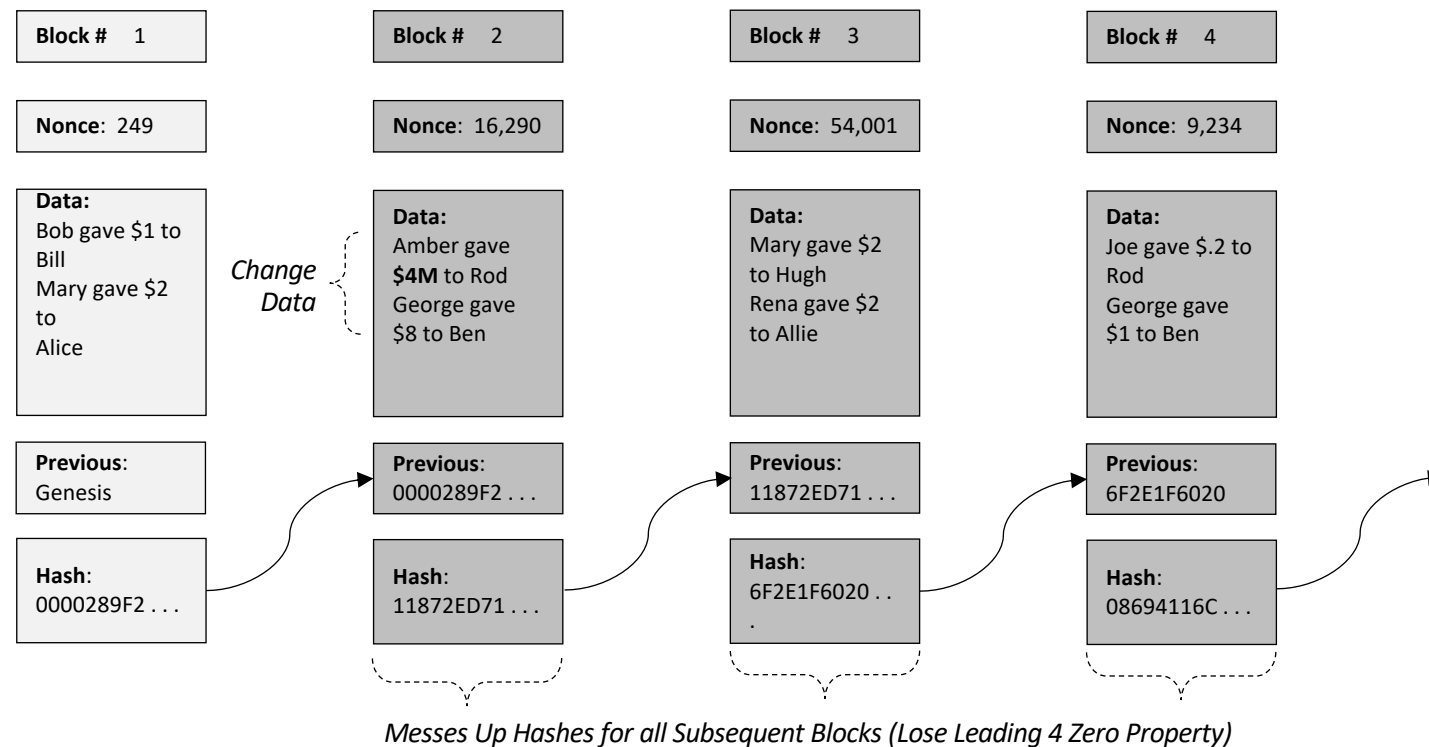
Modern Block Chain Usage



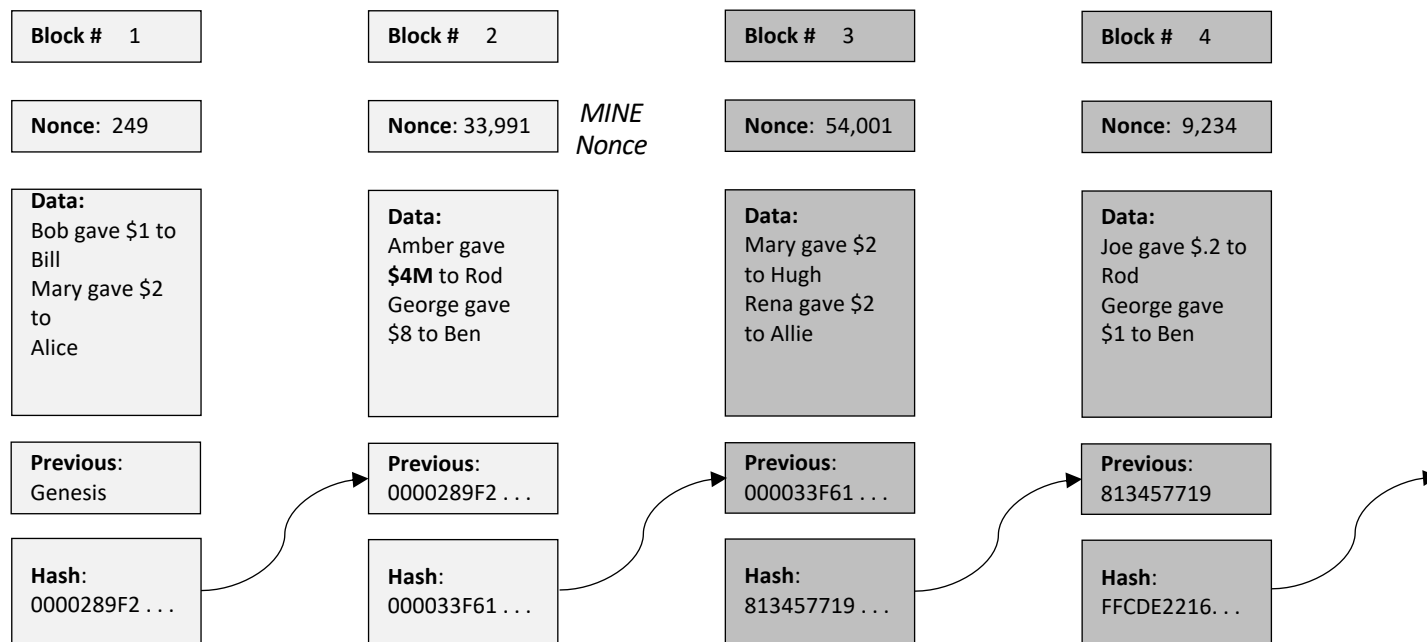
Modern Block Chain Usage



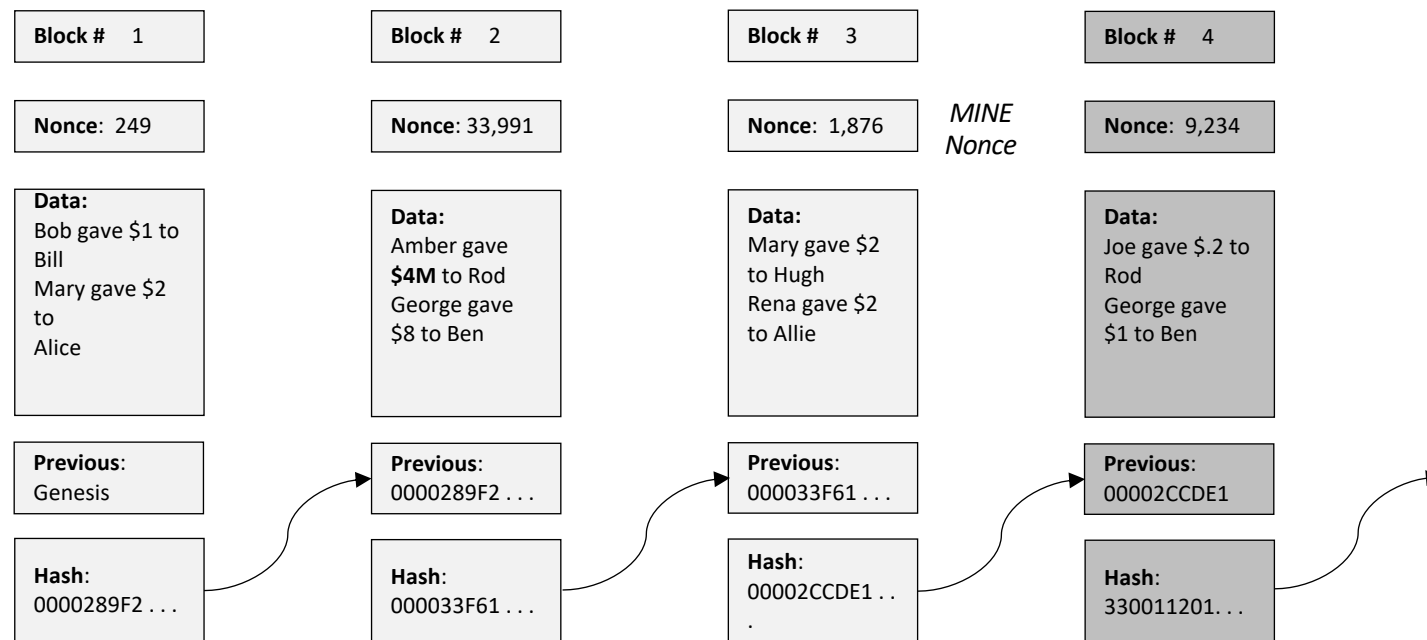
Modern Block Chain Usage



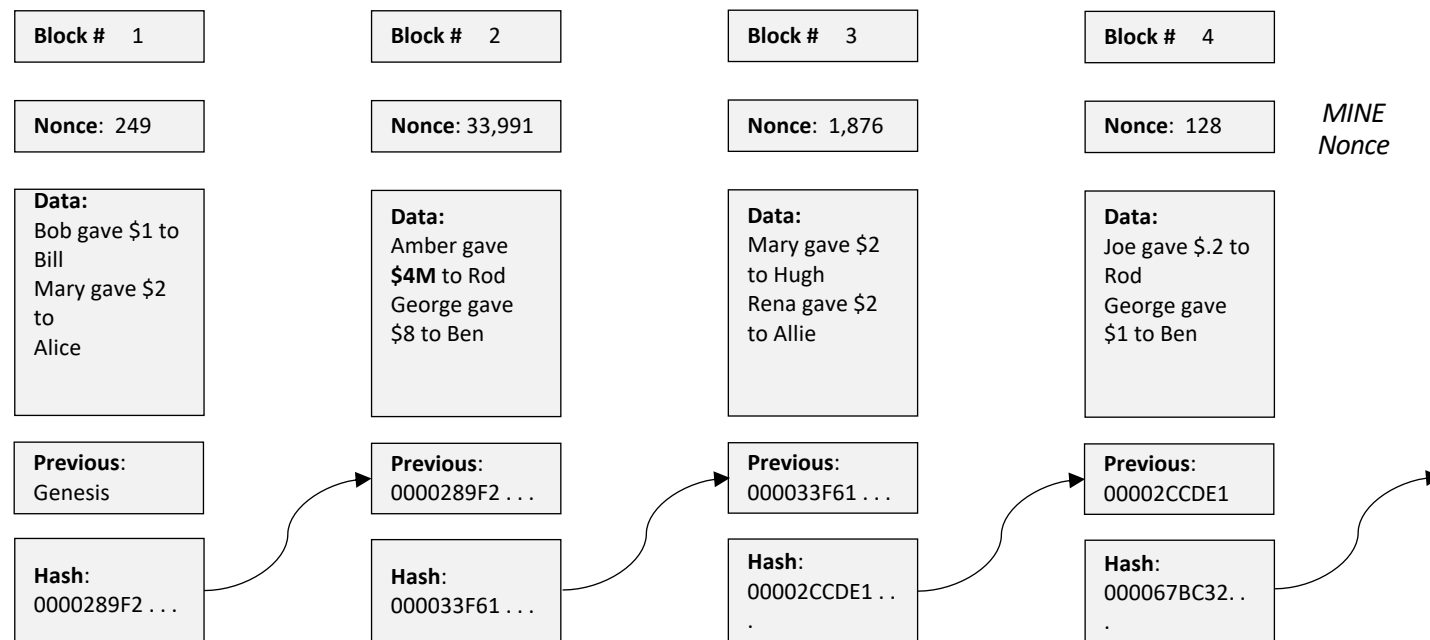
Modern Block Chain Usage



Modern Block Chain Usage



Modern Block Chain Usage



Week 6

