Stu Henderson’s Clear Explanation of Encryption and Digital Certificates

The Henderson Group
50702 Newington Road
Bethesda, MD 20816
(301) 229-7187

Abstract

In this session, we will explain how digital certificates work in terms that you can understand.   We will show you how to apply this understanding to develop policy and practices for PKI (Public Key Infrastructure).

We will show you some examples of certificate usage by showing some control statements that invoke SSL for FTP, MQ or DB2, and briefly cover, how certificates are protected through RACF.

AGENDA

I Introduction and Types of Encryption

II I Just Inherited a Potfull of Someone Else’s Digital Certificates

III PKI (Public Key Infrastructure)

IV Practical Application: SSL with FTP, DB2 and MQ

V Summary and Call to Action

I Introduction and Types of Encryption

- What Encryption is Not: The Cocktail Party
- Symmetric & Asymmetric keys
- Digital certificates
- Hardware / Software

I Introduction and Types of Encryption

- Imagine Sam at a Crowded Cocktail Party. He Sees an Attractive Woman Named Janet on the Other Side of the Room. He'd Like to Have Lunch with Her. Just Her.

- He Could Write an Invitation on a Cocktail Napkin and Ask Fred, Standing Next to Him, to Pass It On.

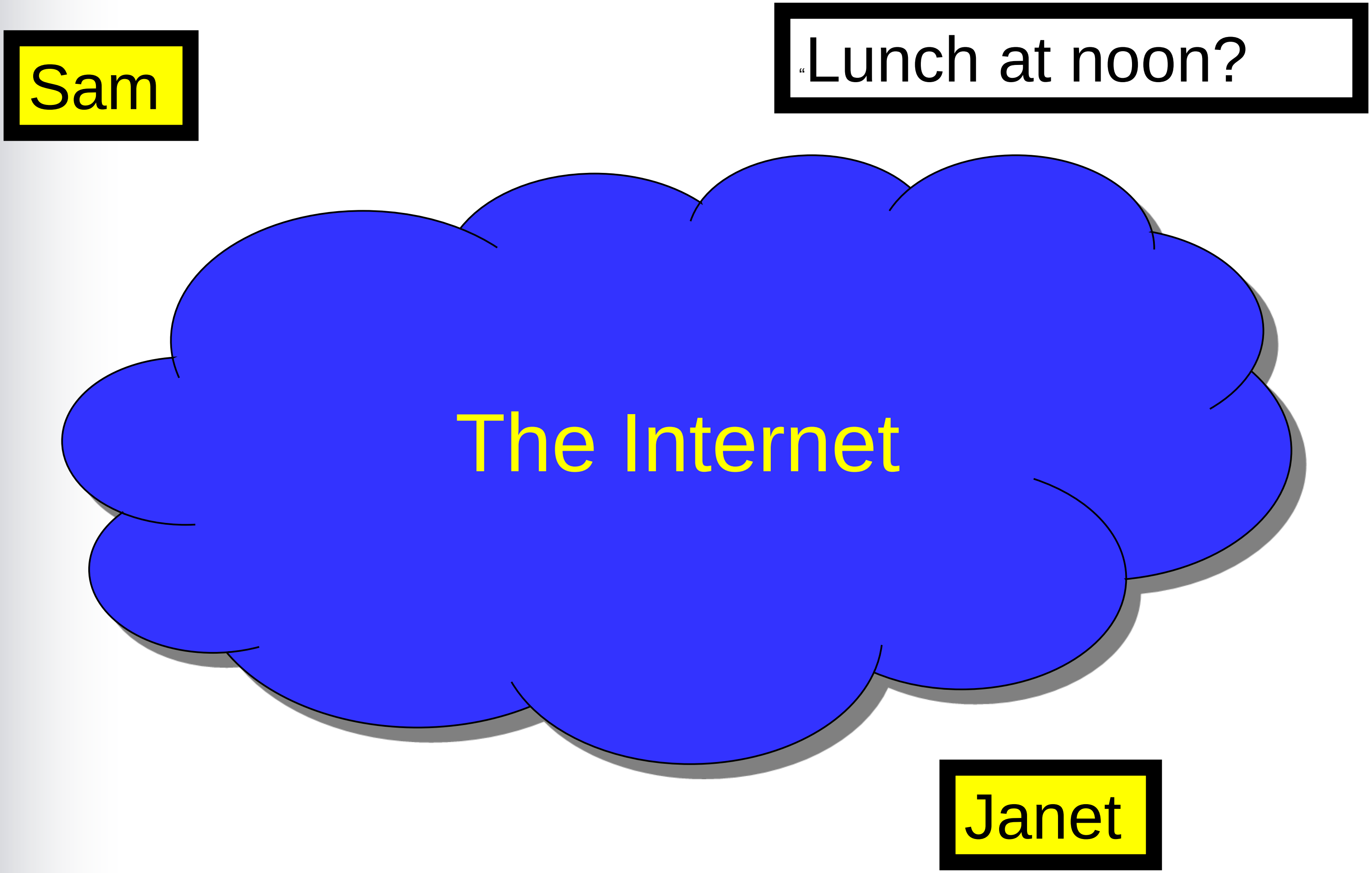
I Introduction and Types of Encryption

- The Risk Assessment Here is Not Difficult.
- Fred is a Self-Centered Individual with a Loud Voice Who Talks with his Mouth Full of Food.
- His Hobby is Avian Topiary.

I Introduction and Types of Encryption

- **So We’ll Show You First How Sam Might Pass His Invitation to Janet Without Encryption.**
- **You Understand of Course that the Crowded Room is Like the Internet, Full of Uncouth People Who Want to Read Others’ Emails.**

Not Encryption



Sam

“Lunch at noon?

The Internet

Janet

Suppose Sam Doesn’t Want Anyone Else to Read the Message

- **He could replace each letter in the message with the letter 3 letters later in the alphabet.**
- **So, *L* would become *O*, and**
  ***u* would become *x*, and so on.**
- **A letter shifter program could do the work for him.**

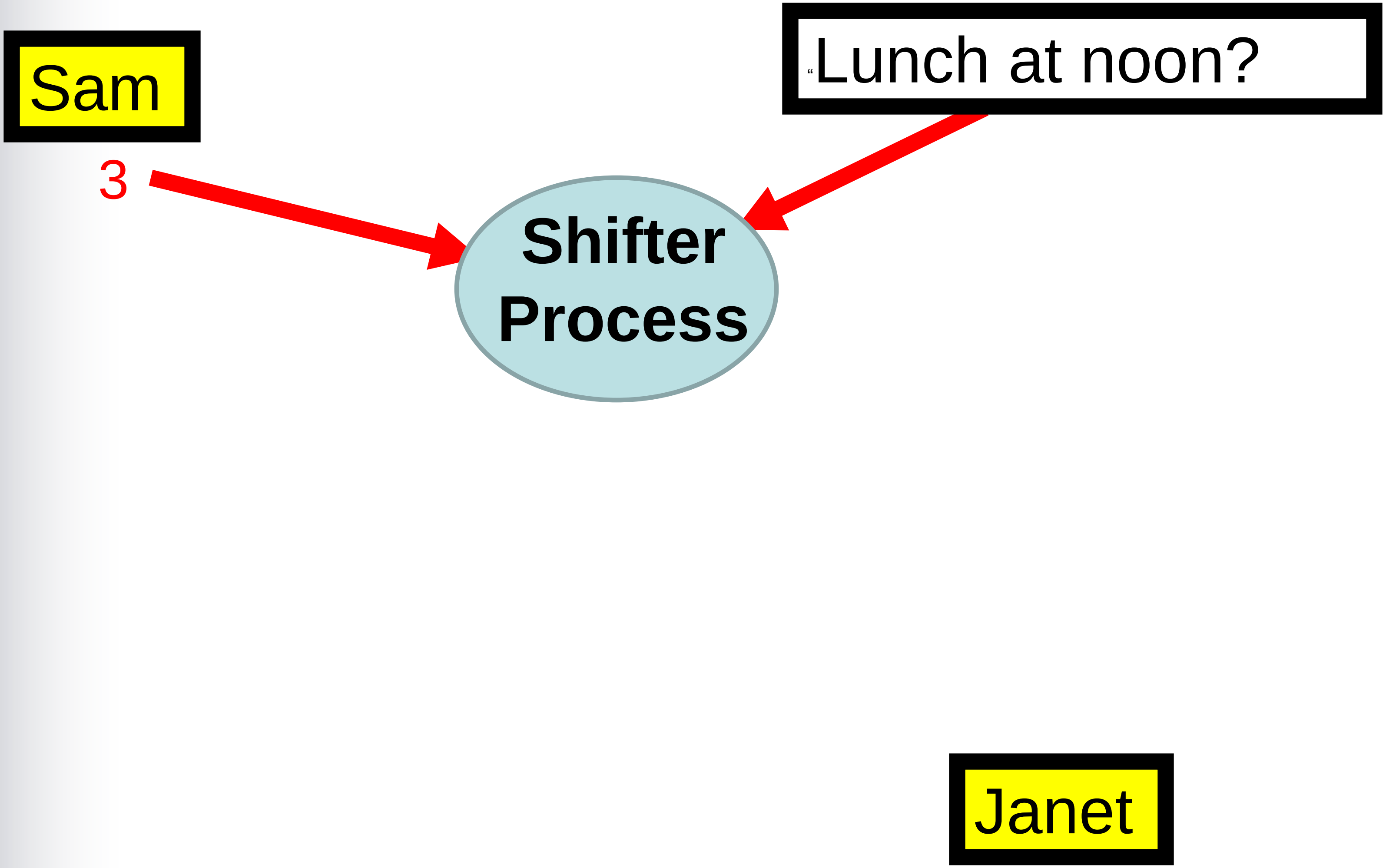
Hiding the Message (Not Encryption)

Sam

“Lunch at noon?

Janet

Hiding the Message (Not Encryption)



Sam

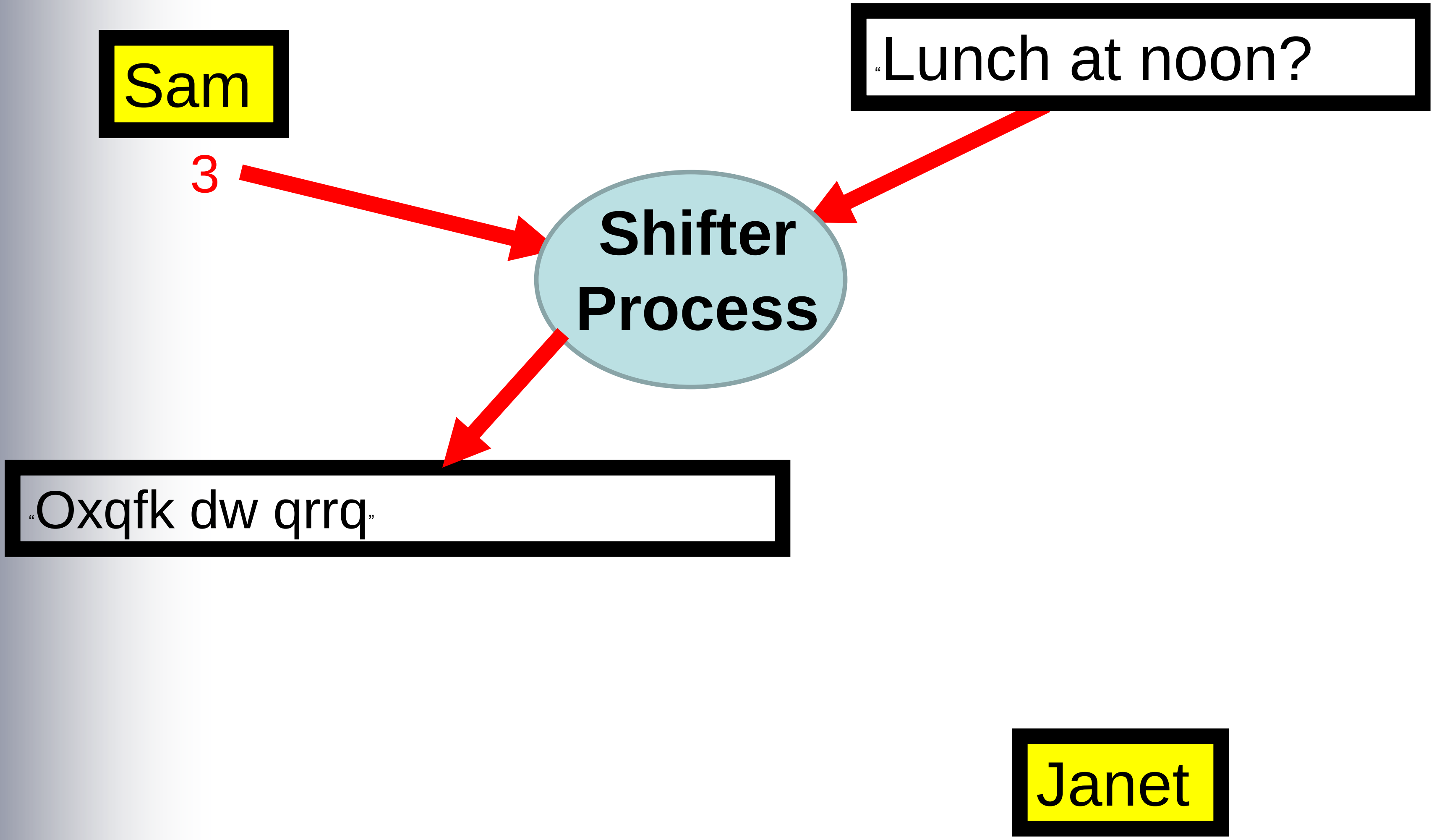
3

“Lunch at noon?

Shifter Process

Janet

Hiding the Message (Not Encryption)



Sam

3

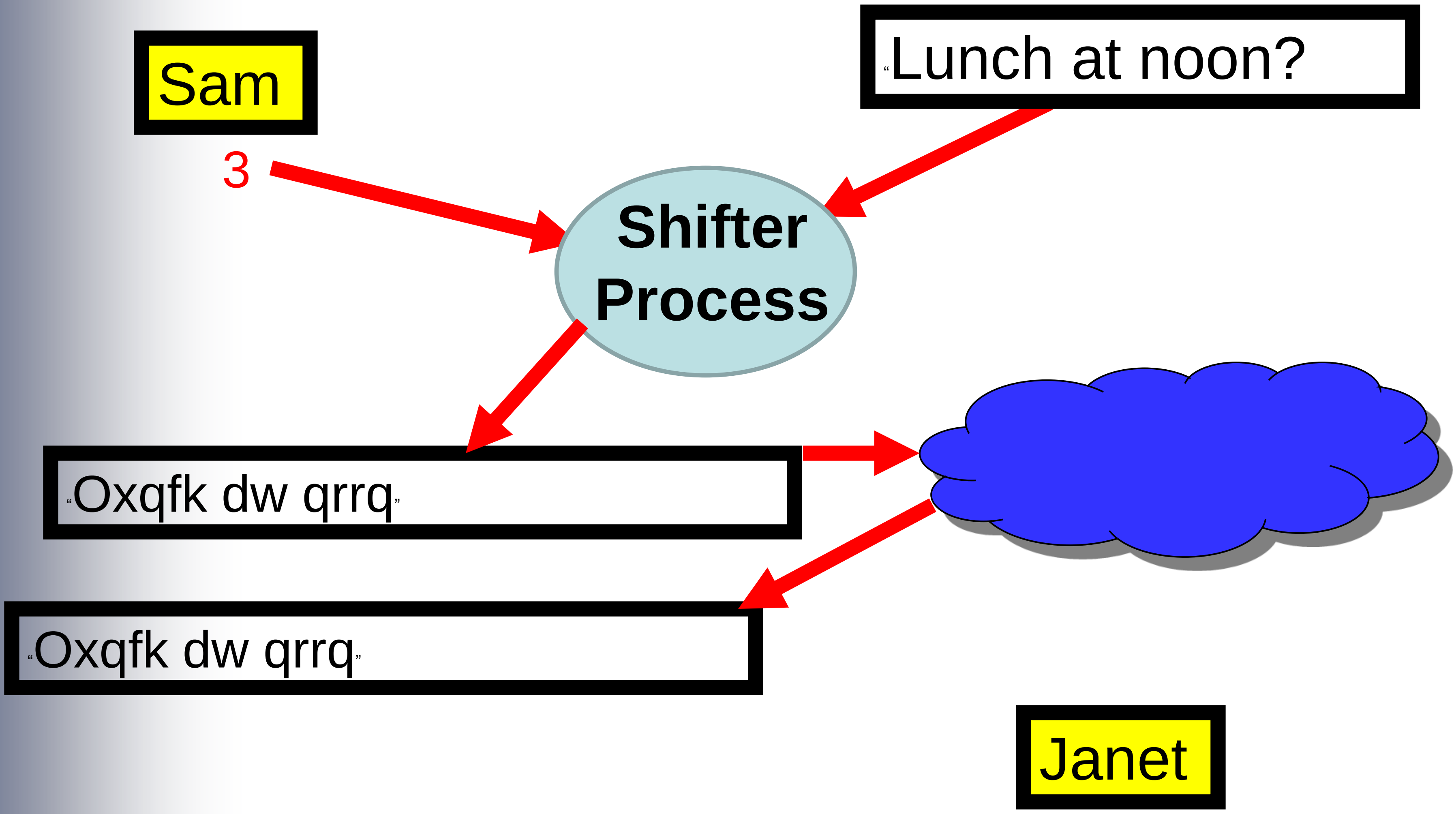
"Lunch at noon?

Shifter Process

"Oxqfk dw qrrq"

Janet

Hiding the Message (Not Encryption)



Sam

3

“Lunch at noon?

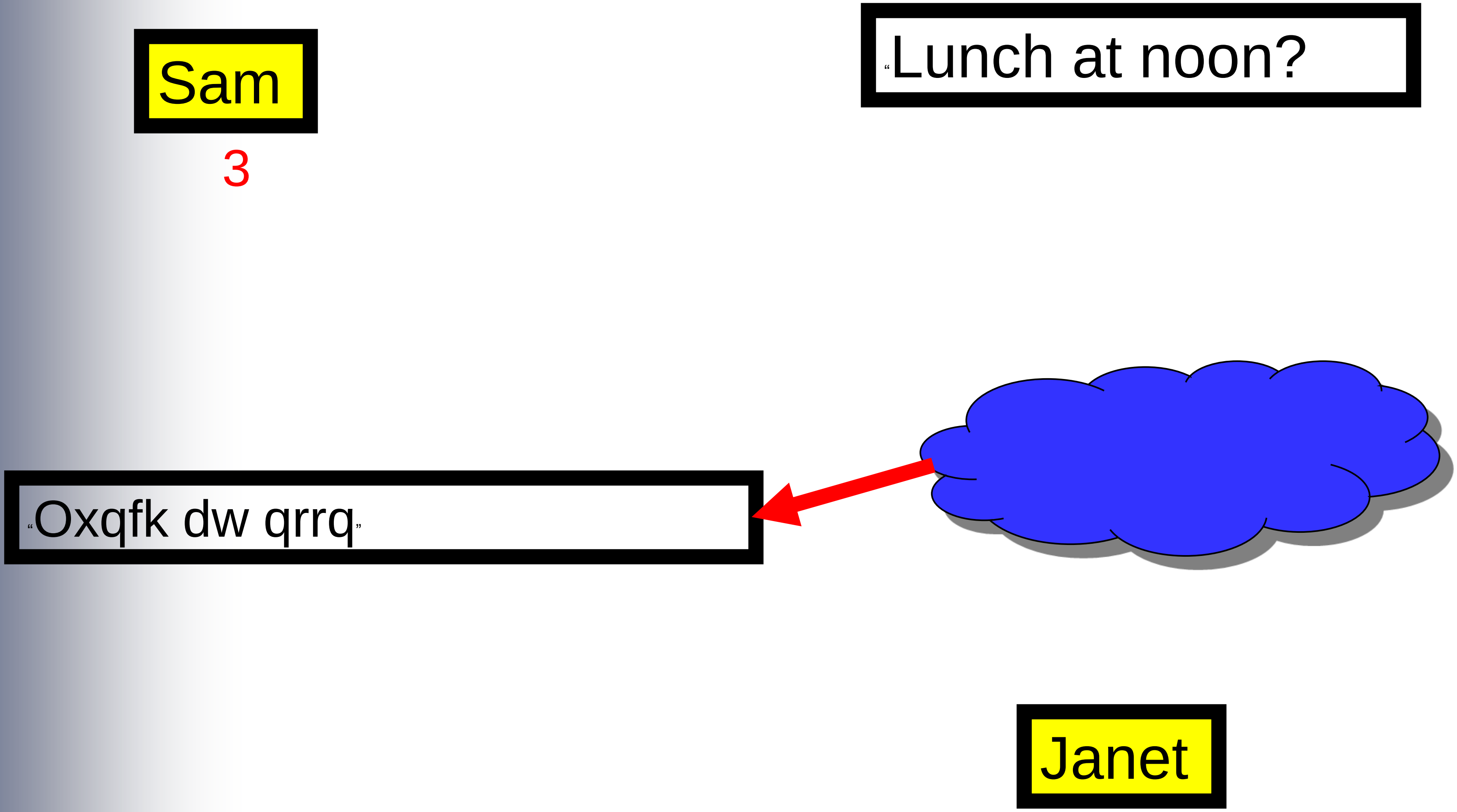
Shifter Process

“Oxqfk dw qrrq”

“Oxqfk dw qrrq”

Janet

Hiding the Message (Not Encryption)



Sam

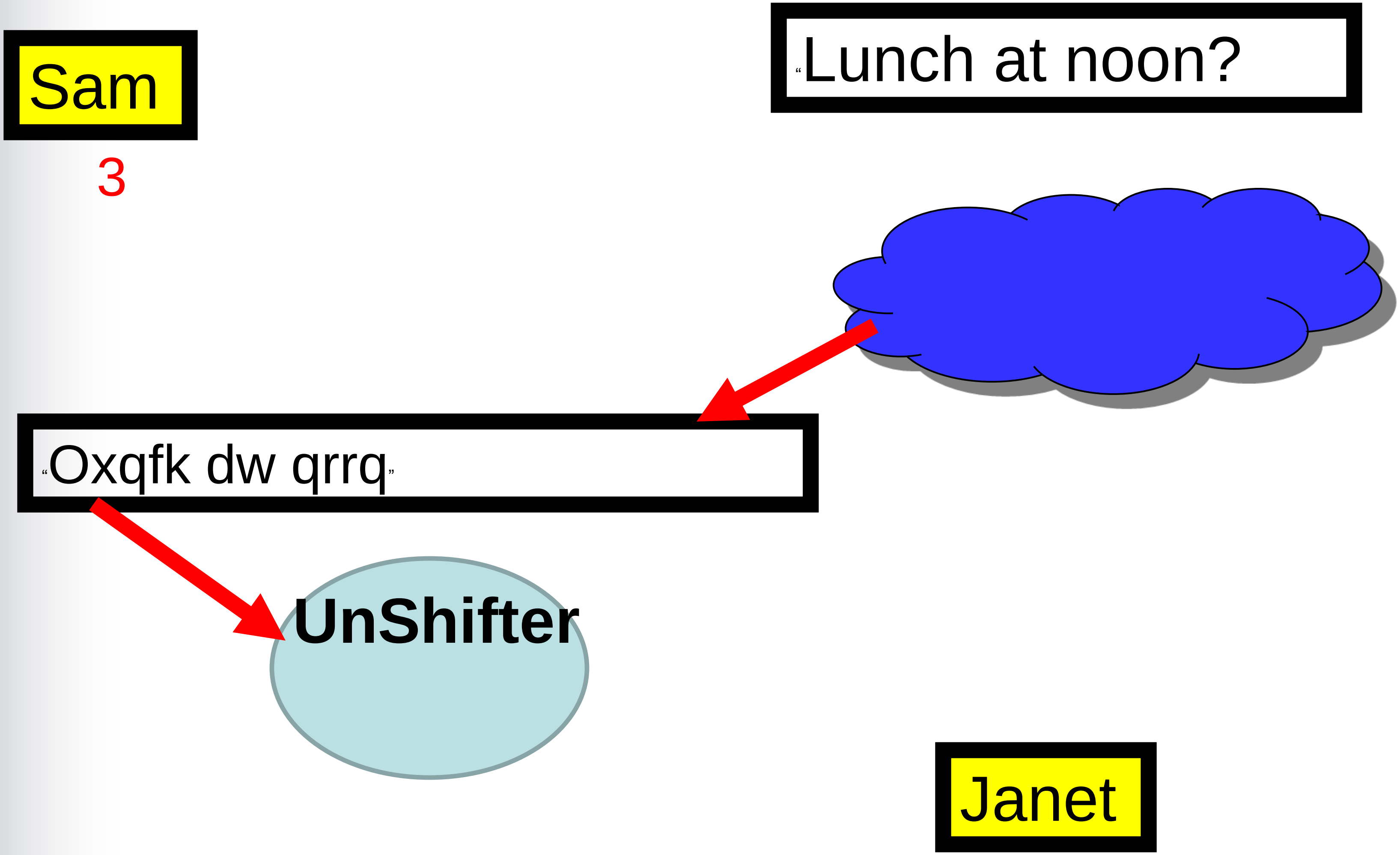
3

“Lunch at noon?

“Oxqfk dw qrrq”

Janet

Hiding the Message (Not Encryption)



Sam

3

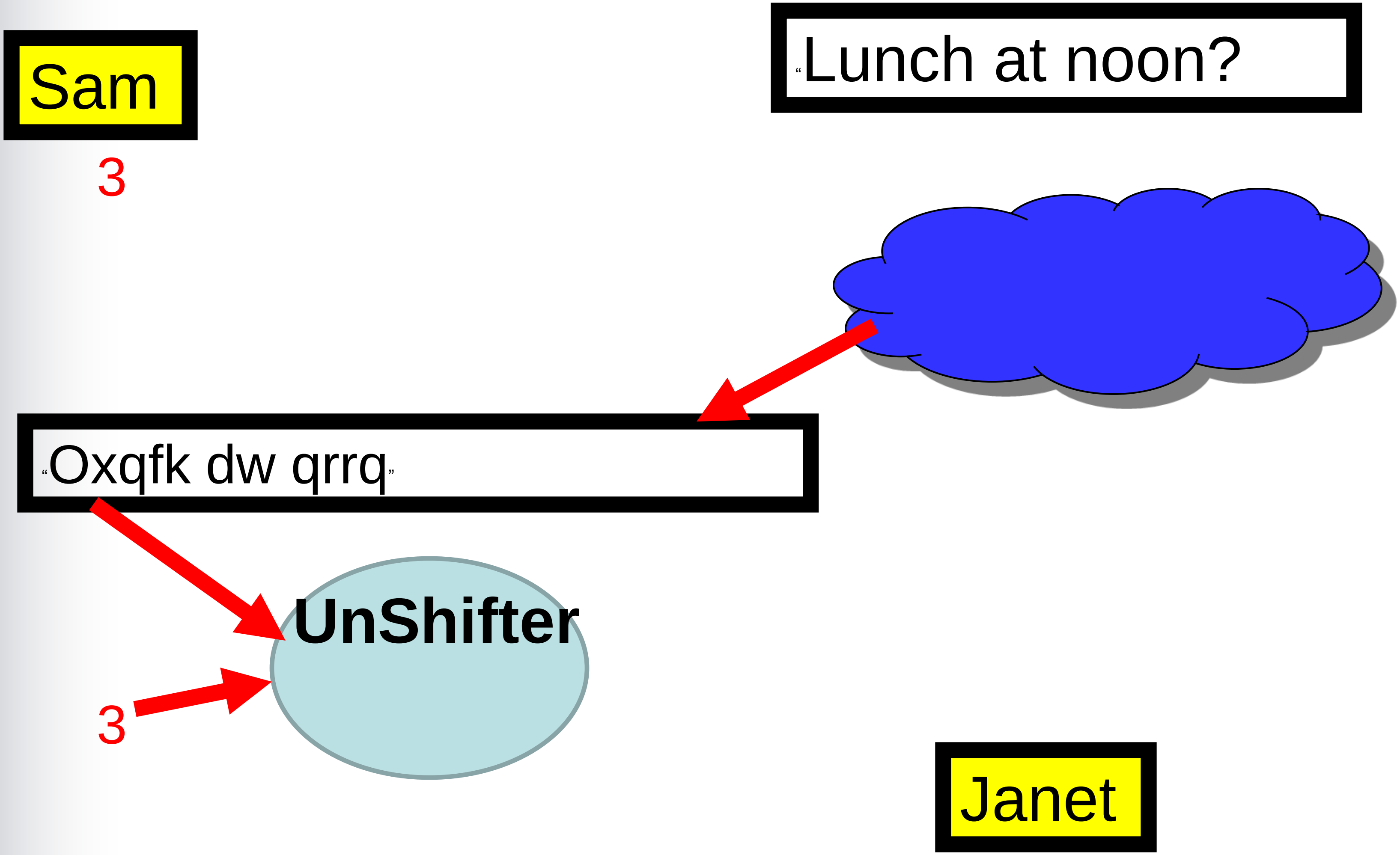
“Lunch at noon?

“Oxqfk dw qrrq”

UnShifter

Janet

Hiding the Message (Not Encryption)



Sam

3

“Lunch at noon?

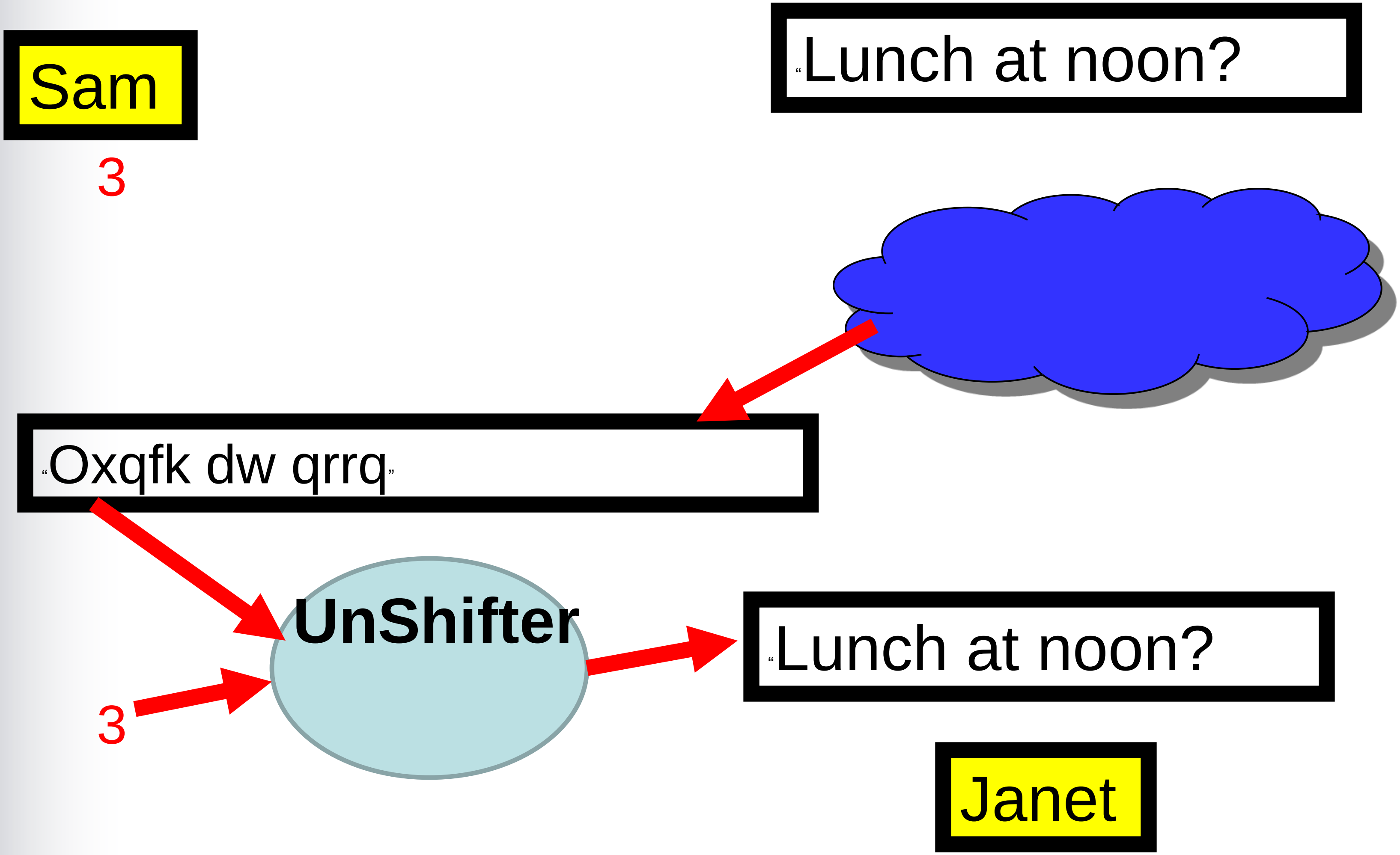
“Oxqfk dw qrrq”

UnShifter

3

Janet

Hiding the Message (Not Encryption)



Sam

3

“Lunch at noon?

“Oxqfk dw qrrq”

UnShifter

3

“Lunch at noon?

Janet

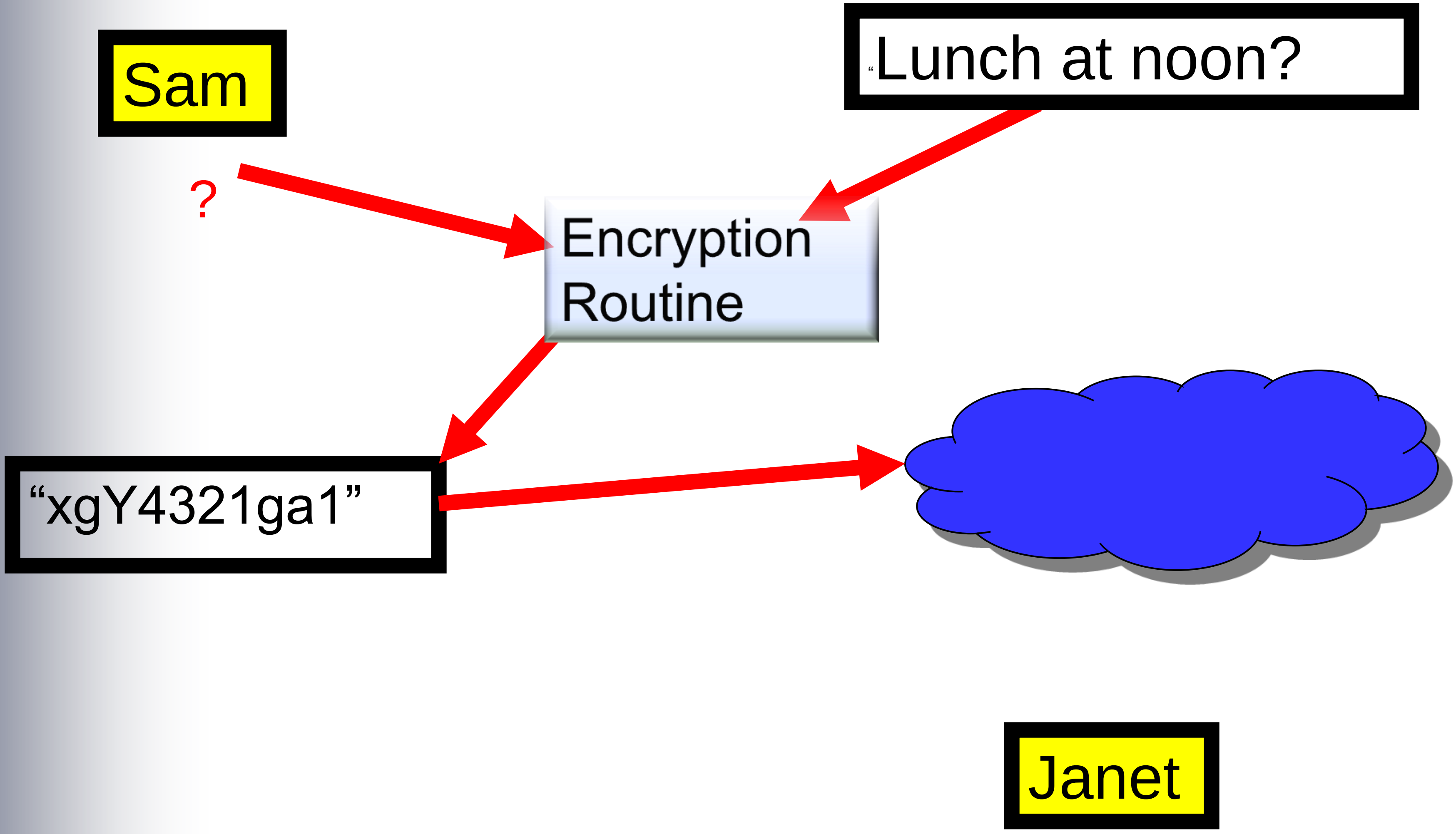
I Introduction and Types of Encryption

- Both Sam and Janet Need to Know That the Number is 3. This is Called the Key.

- But Letter-Shifting is Not Secure, and Can Be Cracked Easily.

- So We Need True, Mathematically Rigorous Encryption.

Symmetric Encryption (One-Key)



Sam

?

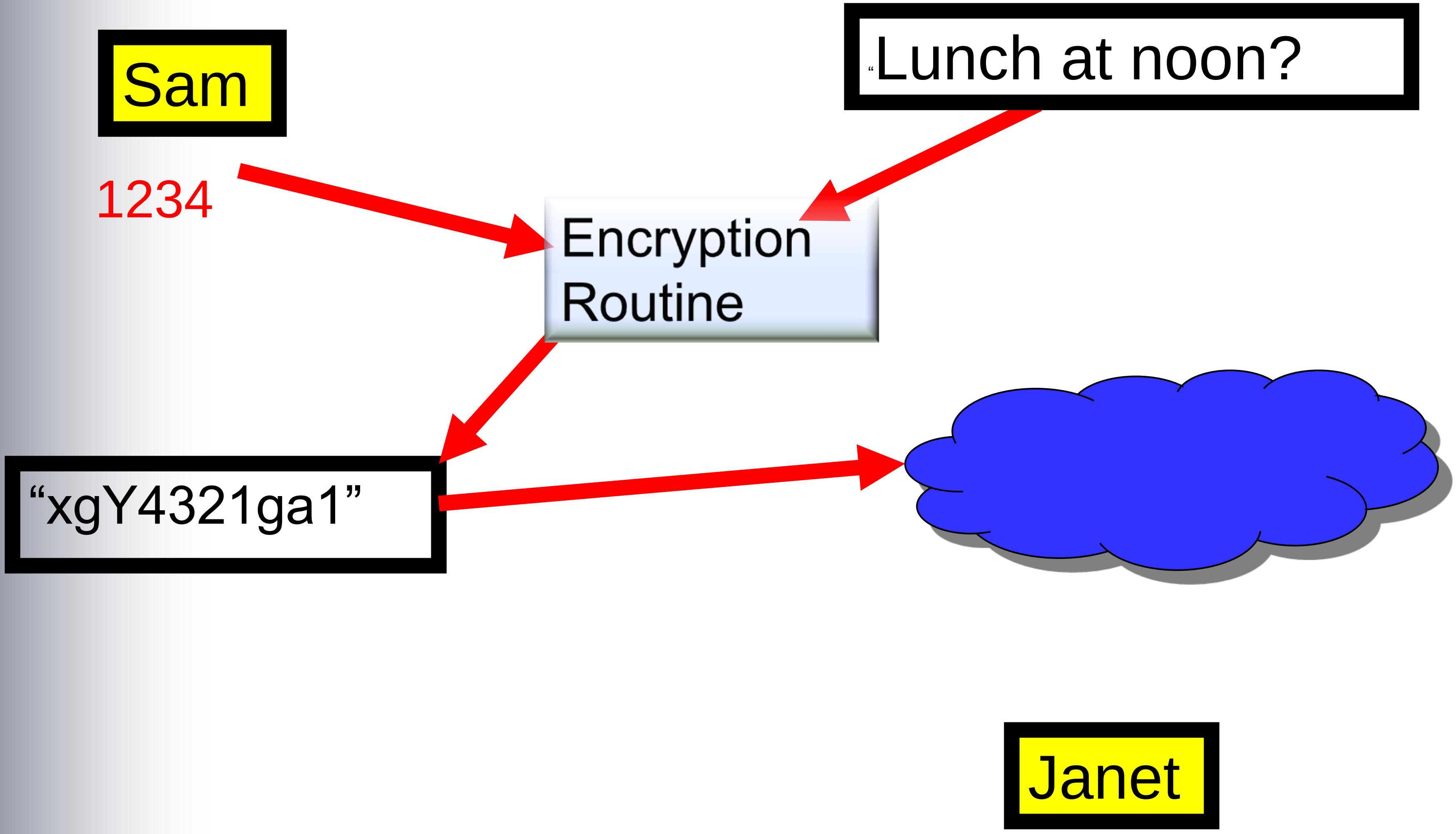
“Lunch at noon?

Encryption Routine

“xgY4321ga1”

Janet

Symmetric Encryption (One-Key)



Sam

1234

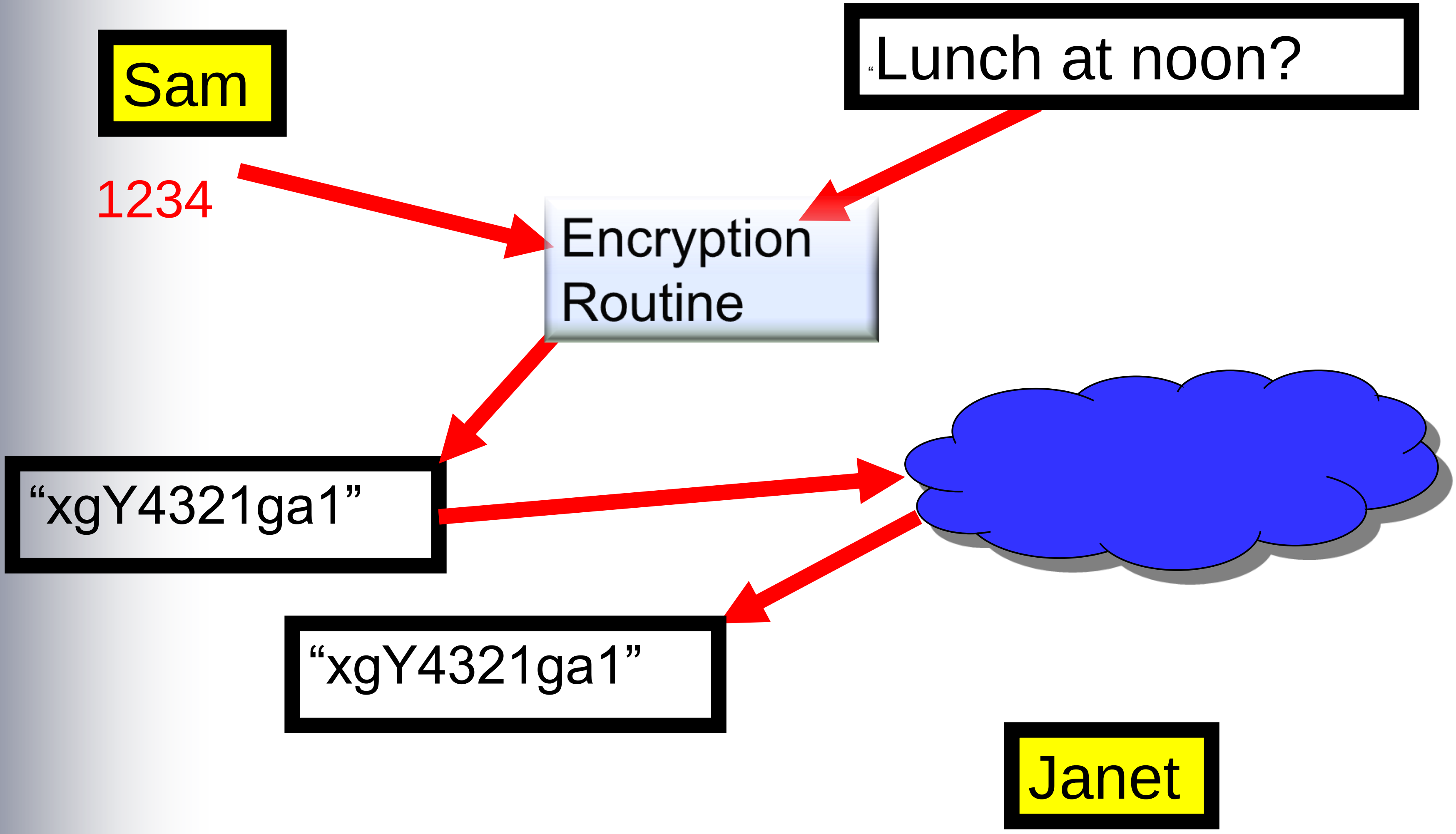
“Lunch at noon?

Encryption Routine

“xgY4321ga1”

Janet

Symmetric Encryption (One-Key)



Sam

“Lunch at noon?

1234

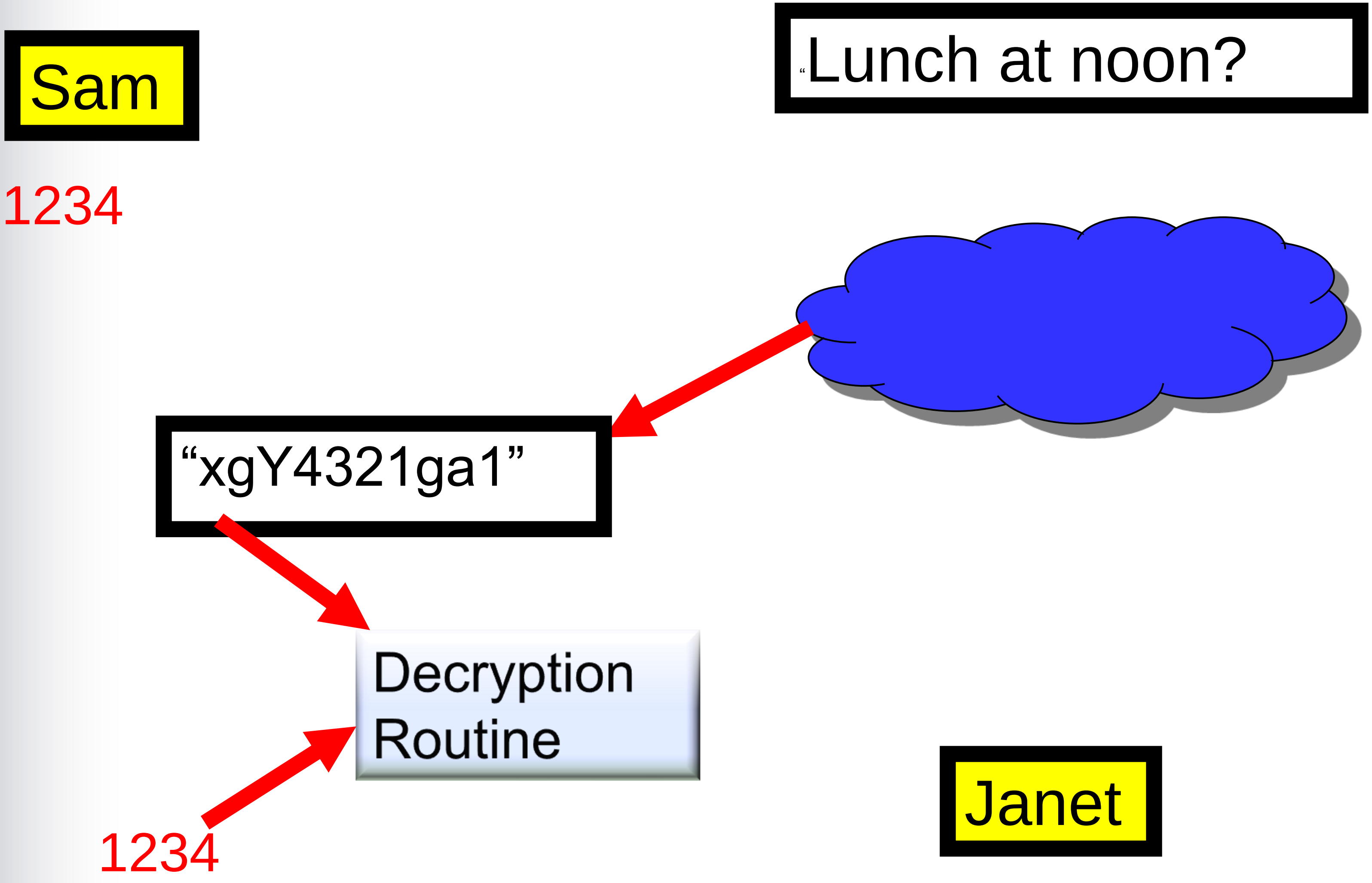
Encryption Routine

“xgY4321ga1”

“xgY4321ga1”

Janet

Symmetric Encryption (One-Key)



Sam

“Lunch at noon?

1234

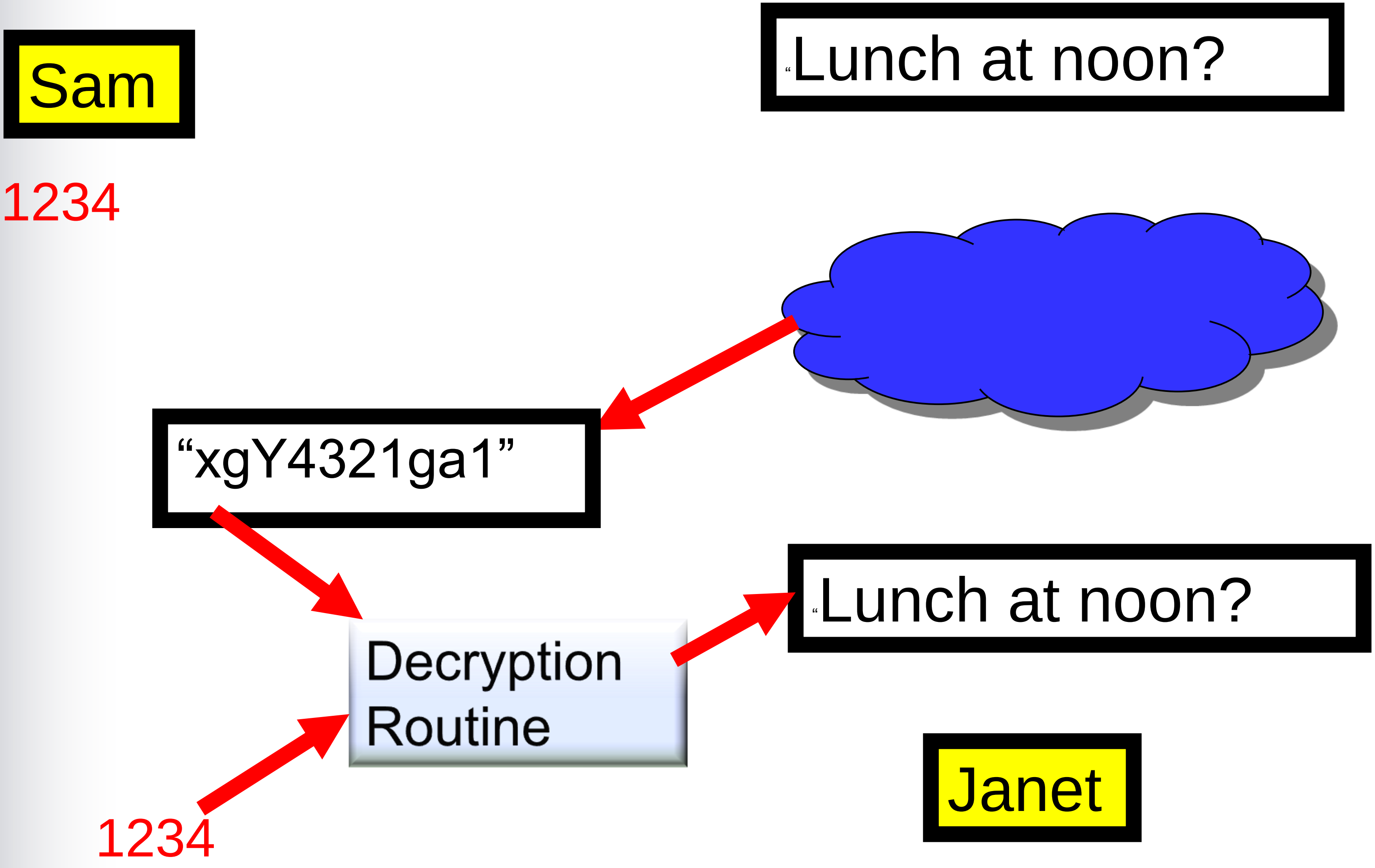
“xgY4321ga1”

Decryption Routine

1234

Janet

Encryption (One-Key)



Sam

“Lunch at noon?

1234

“xgY4321ga1”

Decryption Routine

“Lunch at noon?

Janet

1234

Asymmetric Encryption (Two keys)

- Sam Has a Pair of Keys:
- 5678 (Public)
- 6724 (Private)

- Janet Also Has a Pair of Keys:
- 7734 (Public)
- 4357 (Private)

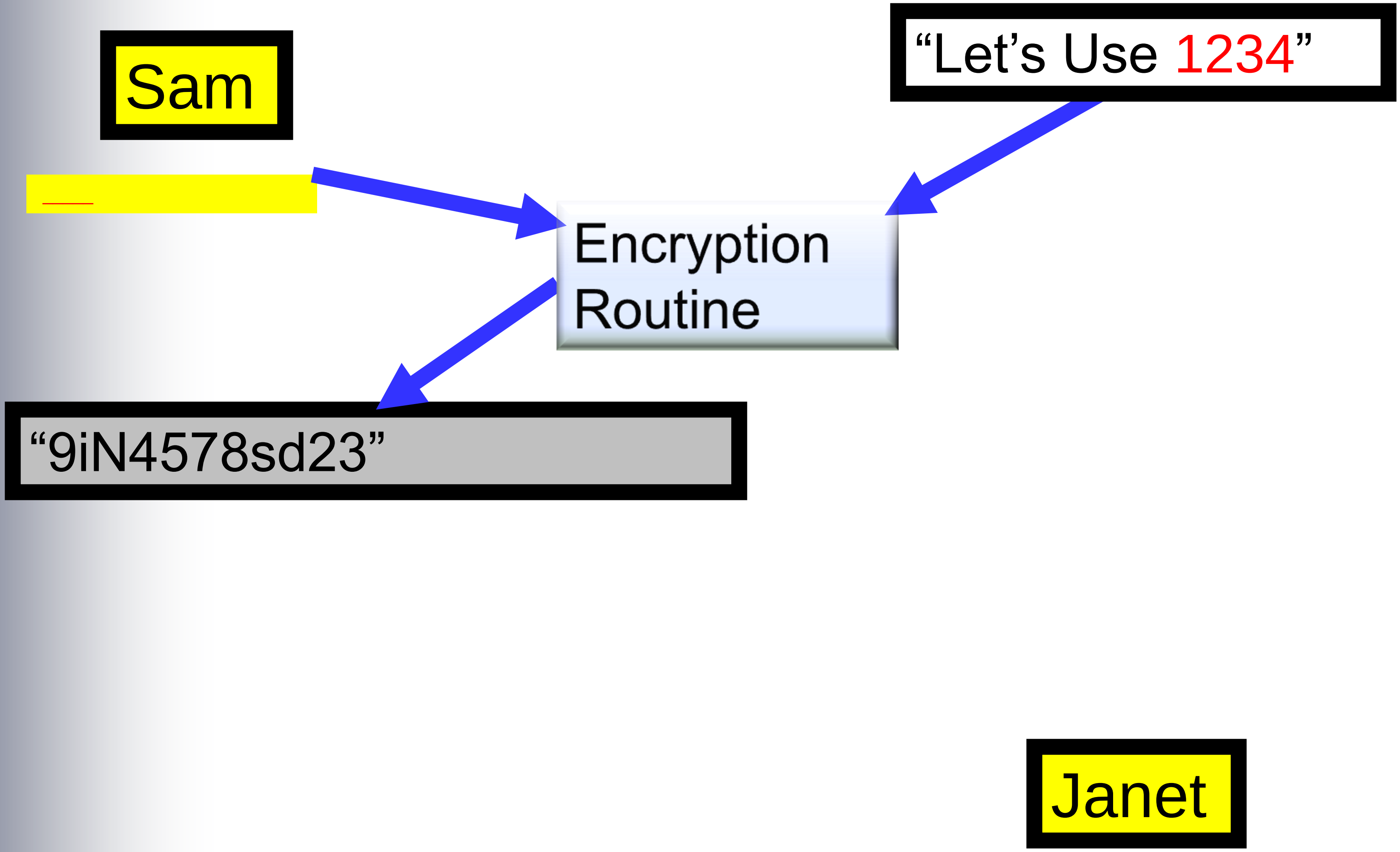
Asymmetric Encryption (Two keys)

Sam

“Let’s Use 1234”

Janet

Asymmetric Encryption (Two keys)



Sam

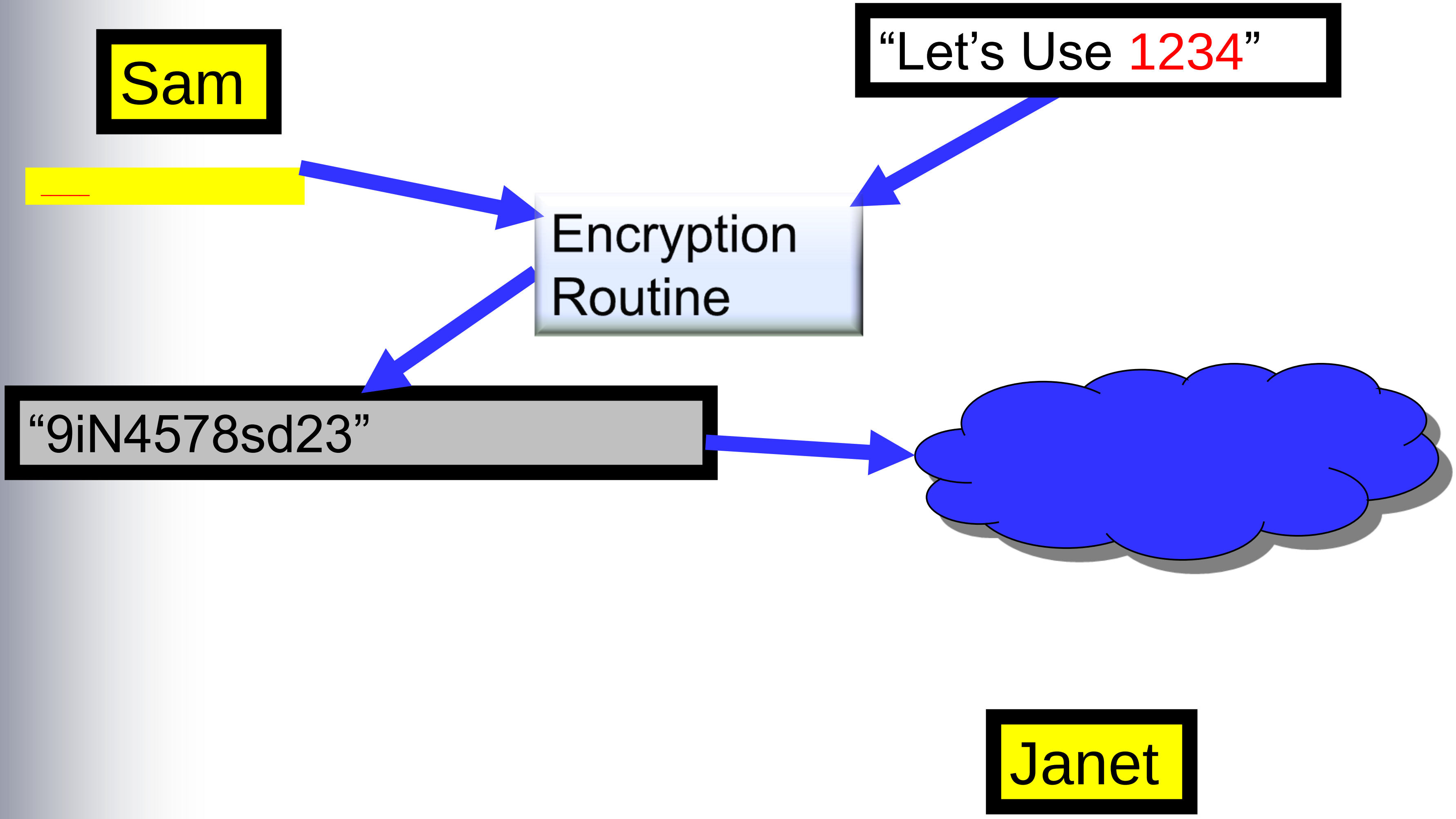
“Let’s Use 1234”

Encryption Routine

“9iN4578sd23”

Janet

Asymmetric Encryption (Two keys)



Sam

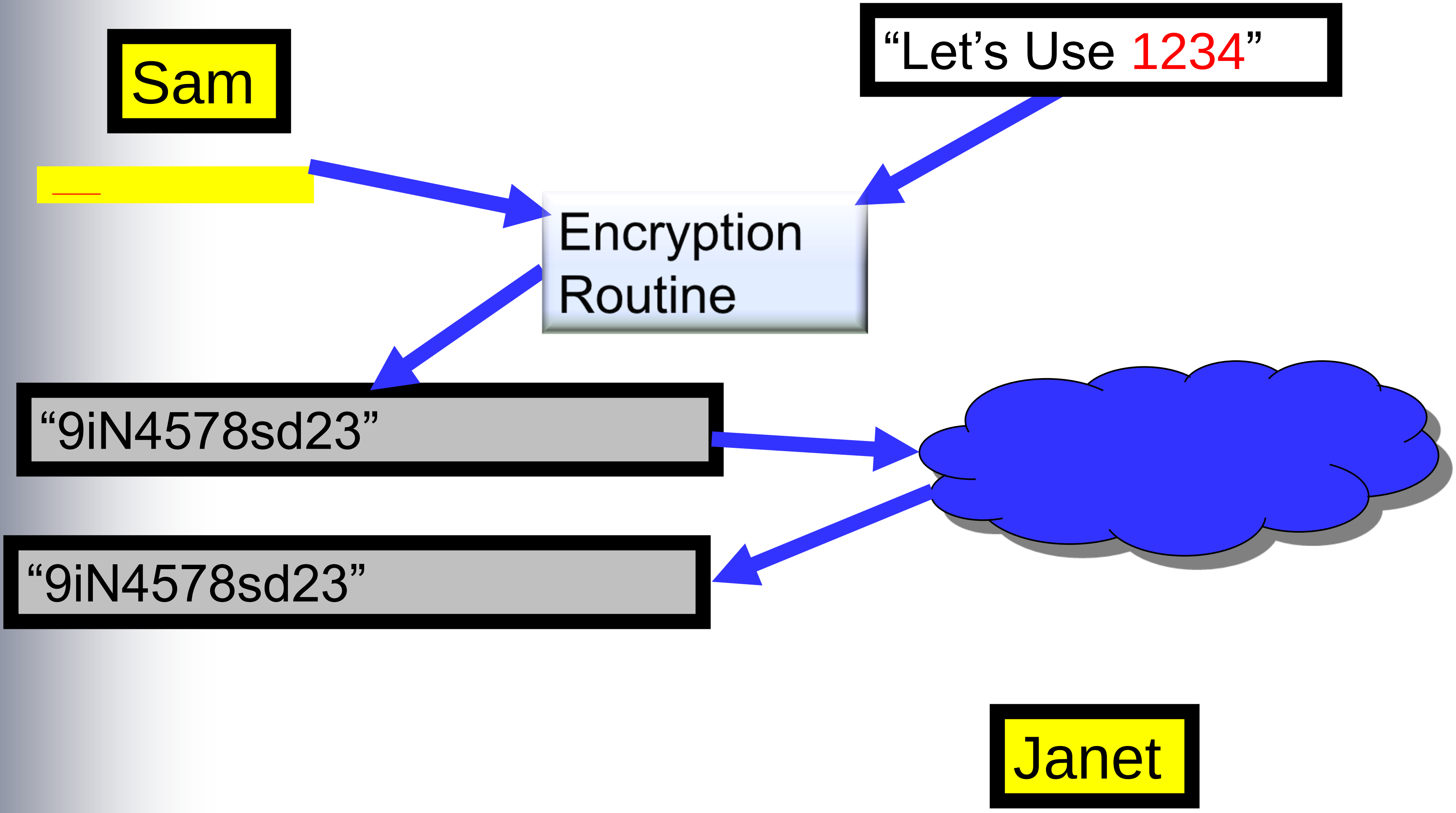
“Let’s Use 1234”

Encryption Routine

“9iN4578sd23”

Janet

Asymmetric Encryption (Two keys)



Sam

“Let’s Use 1234”

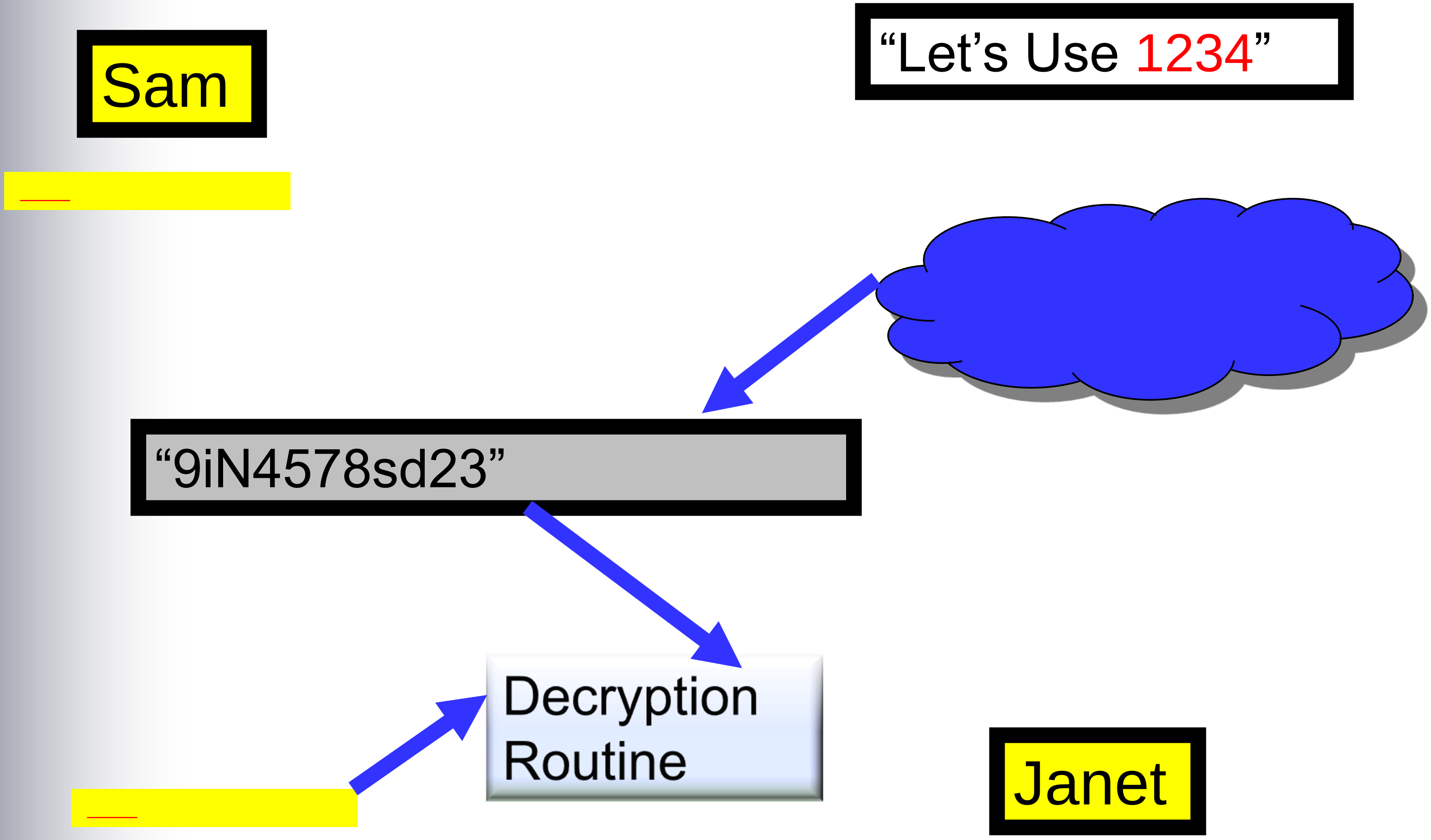
Encryption Routine

“9iN4578sd23”

“9iN4578sd23”

Janet

Asymmetric Encryption (Two keys)



Sam

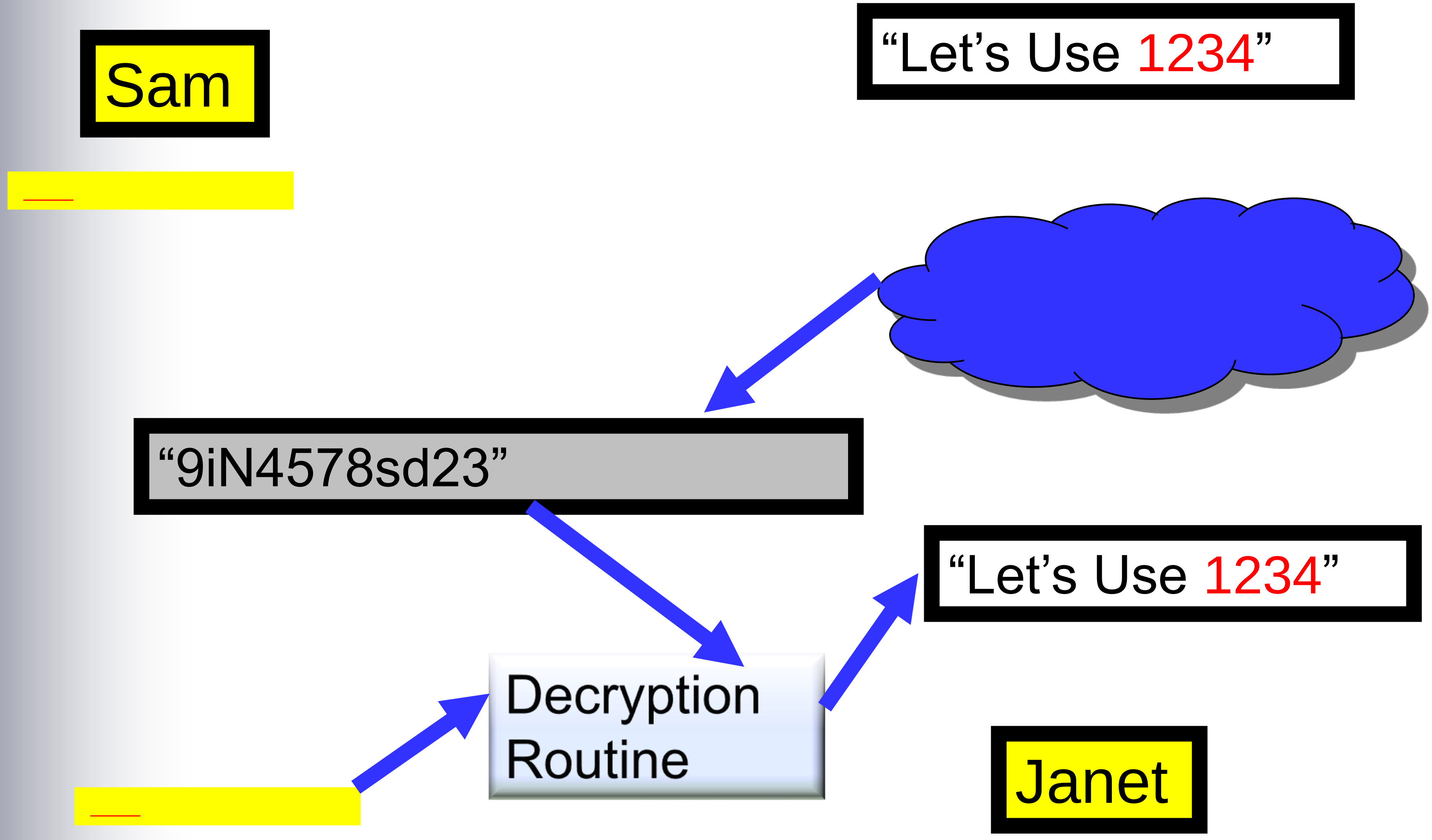
“Let’s Use 1234”

“9iN4578sd23”

Decryption Routine

Janet

Asymmetric Encryption (Two keys)



Sam

“Let’s Use 1234”

“9iN4578sd23”

“Let’s Use 1234”

Decryption Routine

Janet

Asymmetric Encryption (Two keys)

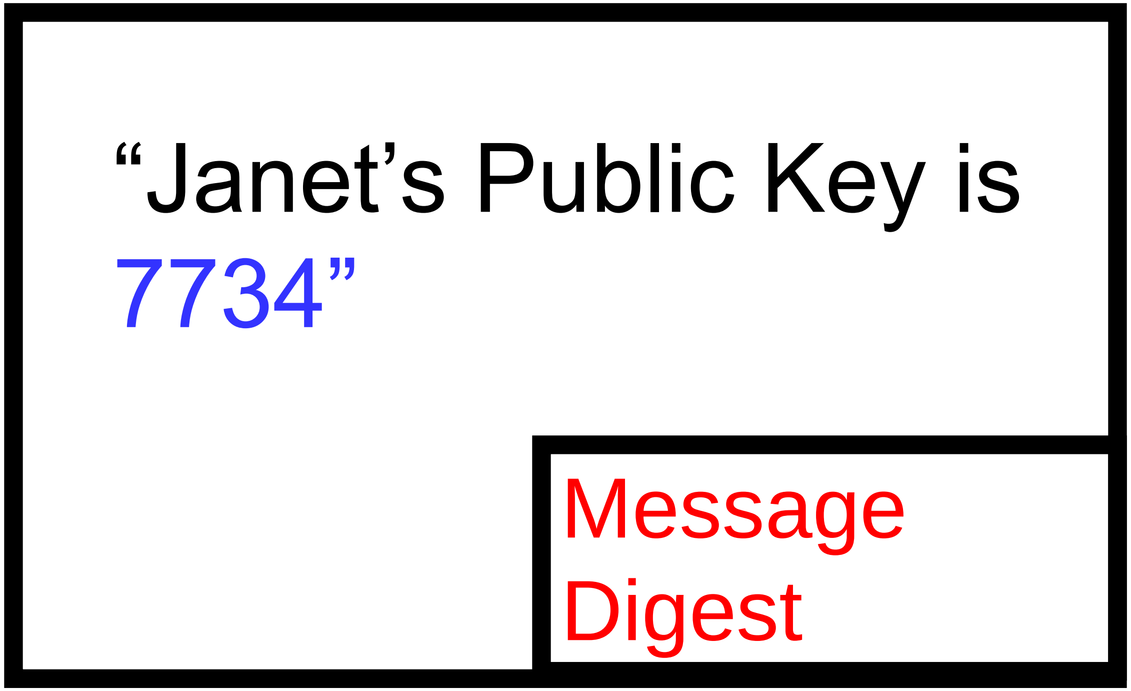
- Sam Has a Pair of Keys:
- 5678 (Public)
- 6724 (Private)

- Janet Also Has a Pair of Keys:
- 7734 (Public)
- 4357 (Private)

Digital Certificate

“Janet’s Public Key is 7734”

MD is a hash of the message encrypted With the sender's private key



"Janet's Public Key is 7734"

Message Digest

I Introduction and Types of Encryption

- **The Whole Business of Using 2 Key Crypto to Set Up 1 Key Crypto is Called SSL (Secure Sockets Layer) Which is Being Succeeded by TLS (Transport Layer Security)**

- **It's What You Use to Enter Credit Card Info Over the Internet**

I Introduction and Types of Encryption

- **SSL and TLS Are Built Into Internet Explorer and Every Other Browser.**

- **Also in RACF, ACF2, TopSecret AND TCP/IP**

- **When It’s Implemented in Mainframe TCP/IP, It’s Called “System SSL”, and Any Program Can Invoke It**

I Introduction and Types of Encryption

- **Use Symmetric When You Control Both Ends of the Connection (1 Key)**
- **Use Asymmetric (2 Key) When You Only Control One End (For Example, Your Mainframe Is an Internet Server)**
- **If Asymmetric, You Need Digital Certificates, Perhaps Stored in RACF, ACF2, or TSS**

I Introduction and Types of Encryption

- **If Asymmetric, Decide Which of These Functions You Need (In Order to Know How Many Sets of Keys):**
  - Provide Encryption
  - ID the Server
  - ID the Client
  - Digital Signature (not Certificate!)
  - Non-Repudiation (Proof Not Altered in Transit, Proof of Who Sent the Message)
- **Which of These Needs Keys for Just the Server, Which for Client Too?**

II I Just Inherited a Potfull of Someone Else's Digital Certificates

- What's Really in a DigCert
- How to Organize and Maintain the Mess

II I Just Inherited a Potfull of Someone Else's Digital Certificates

- DigCerts Don't Really Say *"Janet's Public Key is…"*

Instead They Use LDAP Names: ***"the Public Key of:***

c=US o=HendersonGroup

ou=WestDiv

ou=SalesDept cn= Janet

is …"

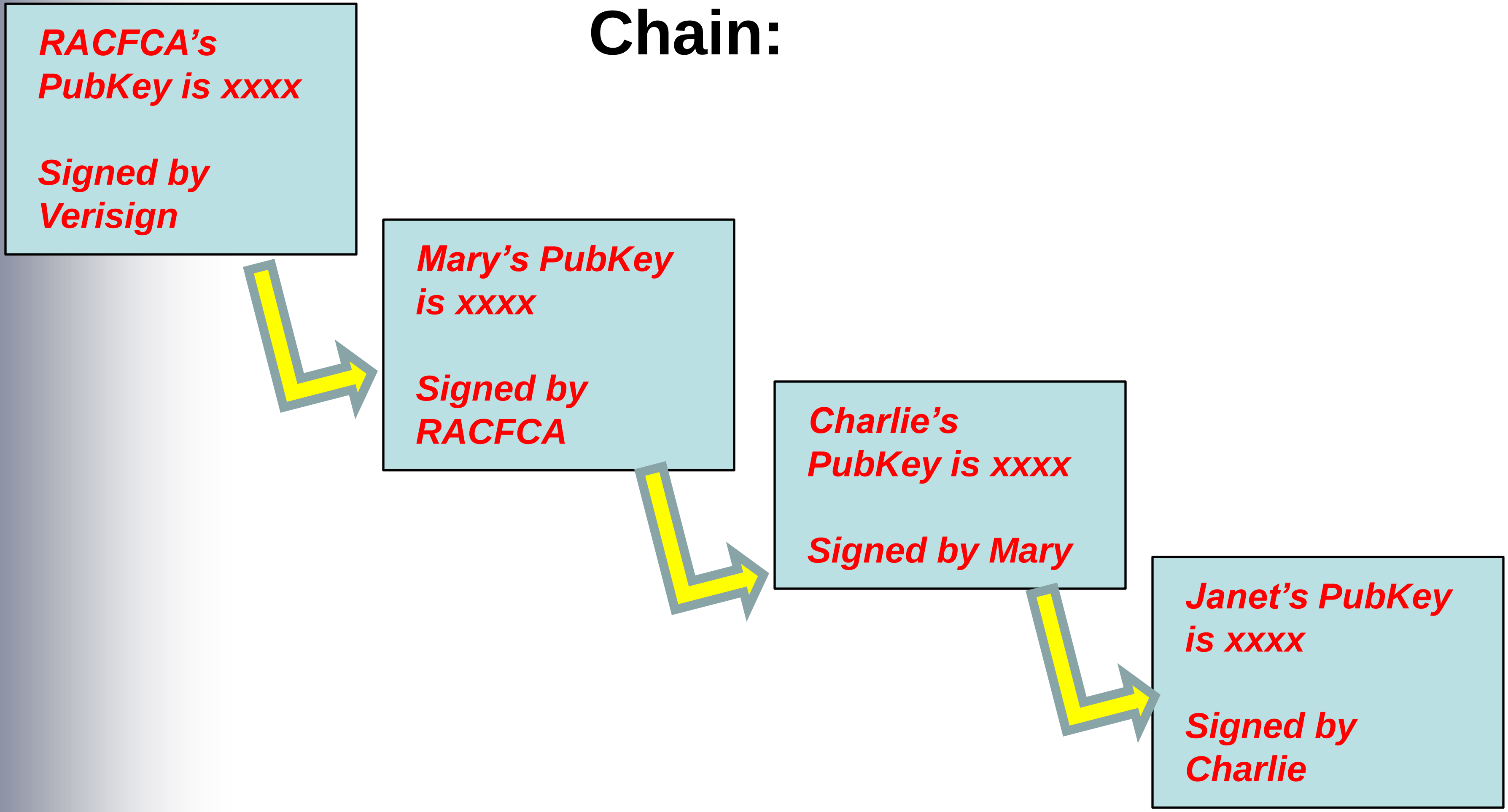
- If You've Ever Administered Active Directory, You Know These Names

II I Just Inherited a Potfull of Someone Else's Digital Certificates

- DigCerts Also Have:
  - – LDAP Name of the Signer
  - – Expiration Dates
  - – TRUST or NOTRUST Indicator

II I Just Inherited a Potfull of Someone Else’s Digital Certificates

- With Signer’s Name They Can Make a Chain:



RACFCA’s PubKey is xxxx

Signed by Verisign

Mary’s PubKey is xxxx

Signed by RACFCA

Charlie’s PubKey is xxxx

Signed by Mary

Janet’s PubKey is xxxx

Signed by Charlie

II I Just Inherited a Potfull of Someone Else’s Digital Certificates

- But All These Loose DigCerts Are Too Difficult to Specify
- So I Put Them All on a KeyRing, and Associate the Name of the KeyRing with Janet’s RACF Userid
- And Now I Just Specify the KeyRing Name (For Example in the Control File for FTP)

II I Just Inherited a Potfull of Someone Else’s Digital Certificates

- So To Organize My Mess, I Sort Out All the Chains of One DigCert Signing Another

- And I Consider Making a KeyRing for Each

II I Just Inherited a Potfull of Someone Else's Digital Certificates

- DigCerts Also Have A TRUST Flag

Janet's PubKey is xxxx

TRUST / NOTRUST

Signed by Charlie

II I Just Inherited a Potfull of Someone Else's Digital Certificates

- **So I Sort Out All the DigCerts Marked NOTRUST and Set Them Aside, Because They Should Be Irrelevant**

- **Ways to Organize:**
  - **Chains of Authorization and KeyRings**
  - **Set All with NOTRUST on the BackBurner**

II I Just Inherited a Potfull of Someone Else's Digital Certificates

- DigCerts Also Have Expiration Dates:

Janet's PubKey is xxxx

Do Not Eat After July 13, 2022

Signed by Charlie

II I Just Inherited a Potfull of Someone Else’s Digital Certificates

If You Want to Enjoy Bastille Day in Peace, What Administrative Procedures Would Help You?

Janet’s PubKey is xxxx

Do Not Eat After July 13, 2022

Signed by Charlie

II I Just Inherited a Potfull of Someone Else's Digital Certificates

- How to Audit or Self-Assess:
  - Follow the Steps from "Prove Who You Are and Get a Pair of Keys" to "Follow the Chain of Authorization to Verify the Key" to "Set Up Encryption" to "Revoke this Certificate"
  - Ask "Does Each Step Do What It's Supposed to Do?
  - Oh Yeah, Start By Asking Which of the Five Functions It's Supposed to Do

II I Just Inherited a Potfull of Someone Else’s Digital Certificates

- Ways to Organize:
  - – Chains of Authorization and KeyRings
  - – Set All with NOTRUST on the BackBurner
  - – Mark Expiration Dates on Calendar Along with “Remember to Pay Rent”

II I Just Inherited a Potfull of Someone Else's Digital Certificates

- Did You Remember to Keep Track of the Expiration Date for:
  - Janet's DigCert?
  - Charlie's DigCert?
  - Mary's DigCert?
  - RACFCA's DigCert?
  - Verisign's DigCert?

II I Just Inherited a Potfull of Someone Else’s Digital Certificates

- If You Use DigCerts to Prove Who Someone Is, Then What If That Person Terminates Employment?

- (Sort of Like Revoking or Suspending a Userid, But How Different?)

III PKI (Public Key Infrastructure)

- PKI is the Standards, Policy, and Practices You Use to Support Digital Certificates and Public Keys.

- Digital Certificates and Their Keyrings Can Be Stored in the Mainframe Security Software (RACF, ACF2, or TopSecret)

III PKI (Public Key Infrastructure) Digital Certificate Best Practices:

Have Policy and Standards and Procedures for Each Item Listed Below. For each digital certificate, know and document the following:

- **Its Purpose (Encryption, ID Server, ID Client, Digital Signature, Non-Repudiation)**
- **Who Authorized It**
- **Its Chain of Authorization with Other Certificates**

III PKI (Public Key Infrastructure) Digital Certificate Best Practices:

- How and When It was: Authorized, Created, Refreshed, Deleted, Checked (Including Certificate Revocation List)

- Who Is Responsible for Each of the Above

- Legal Obligations and Agreements

III PKI (Public Key Infrastructure) Digital Certificate Best Practices:

Naming and Authorization Standards:
Document Standards for:

- **Names (LDAP RDNs, Keyrings)**
- **Registration, Use of Certificate Revocation Lists, Expiration Periods)**
- **Chains of Authorization, Use of Keyrings**

III PKI (Public Key Infrastructure) Digital Certificate Best Practices:

Naming and Authorization Standards:
Document Standards for:

- **What platforms and Applications Involved and How**
- **Coordination with Userid Administration on Other Platforms**

IV Practical Application: SSL with FTP, DB2 and MQ

- SAF Resource Classes: CSFKEYS and CSFSERV
- RACDCERT Command in RACF
- System SSL
- Configuration Statements for Subsystems (DB2, MQ, etc.)

SAF Resource Classes: CSFKEYS and CSFSERV

- **Security Software Rules in the CSFKEYS Resource Class Contain the Keylabels in the CKDS. Access to the Keys Is Protected as with Dataset Rules. The Keylabel is Only a Label and Does not Contain the Key. The Key is in the CKDS.**

- **Rules in the CSFSERV Resource Class Control the Available Cryptographic Calls to ICSF, i.e. CSFENC, Encoding Call.**

Digital Certificates on z/OS

For System SSL to Work, You Need Certificates.

- **Create CA Certificate for Self-signing**
- **Create SITE Certificate Signed by Own CA (RACF Standard Not Industry Standard)**
- **Distribute the Public Certificate of the CA to Use In Verification of the SITE Certificate**

RACDCERT is a RACF Command to

Administer Rules in These Resource Classes:

- ***DIGTCERT*** **– for Digital Certificates**
- ***DIGTRING*** **– for Keyrings (Collections of Digital Certificates Associated with a Userid**
- ***DIGTNMAP*** **– Contains Certificate Name Filters for Mapping**

Note: ACF2 and TSS Have Comparable Functions

To Control Who Can Issue RACDCERT:

Use FACILITY Class Rules With Names Like This:

IRR.DIGTCERT.*function*

Note the Permission Needed:

- **READ = Your Own Certificates**
- **UPDATE = Certificates for Others**
- **CONTROL = For SITE and CERTAUTH certificates**

System SSL

- **SSL/TLS Can Run Natively or Through Application Policy**
- **Policy agent (usually PAGENT - Runs as a Started Task)**
- **PKI Services Needs to be Activated**
- **OCSF - API is used through a httpd (web daemon)**

Once System SSL is Set Up for You, You Can Invoke It In MQ With a Few Control Statements

- **ALTER QMGR SSLKEYR(keyring)**
- **Or (if shared keyring)**
- **ALTER QMGR SSLKEYR(userid/keyring)**

- **ALTER CHANNEL(channelname)**
- **CHLTYPE(SDR)**
- **SSLCIPH(ciphertype i.e. RC4)**

Once System SSL is Set Up for You, You Can Invoke It In FTP With a Few Control Statements

In the FTP Control File, Statements Like:

- SECURE_LOGIN
- SECURE_CTRLCONN
- SECURE_DATACONN
- SECURE_FTP
- KEYRING

DB2

- **DB2 Can Call ICSF**
- **Keywords Specify Which Columns and Rows to Encrypt (Just Within the Given DB2 Subsystem)**
- **Keyword ENCRYPT Will Encrypt With a Password and Can Be Used on Columns, for Example**

Once System SSL is Set Up for You, You Can Invoke It In DB2 With a Few Control Statements

- DSNZPARM has control statements
- Secure Ports Statements Are Defined in The Bootstrap Dataset
- When Passing Data into or out of the DB2 Subsystem From Other Database Facilities, it uses DRDA
- Define the DRDA SECURE PORT to specify incoming SSL port (listening port) to Enable SSL

Once System SSL is Set Up for You, You Can Invoke It In DB2 With a Few Control Statements

- For Distributed Data Facility
- Define DDF, Listener Port for Incoming SQL Calls, Control Statement is SECPORT

V Summary and Call to Action

- If You Are a Security Administrator or an Auditor, You Need to Understand the Digital Certificates and PKI
- This Session Has Shown You How to Think About Them
- We Hope You'll Look at Your TCP/IP Networks to Apply What We've Covered
- Thanks for Your Kind Attention

For More Info:

See articles and back issues of the RACF User News and Mainframe Audit News at www.stuhenderson.com

- IBM manual “z/OS Communications Server: IP Configuration Reference”, SC31-8776
- IBM manual “z/OS Communications Server: IP Configuration Guide”, SC31-8775
- Computer Associates Cookbooks for ACF2 and TopSecret

For More Info: IBM Manuals

- IBM z Server zSeries 990 (z990) Cryptography Implementation SG24-7070-00
- z/OS V1R9.0 Cryptographic Services ICSF System Programmer's Guide SA22-7520-10
- IBM Principles of Operations SA22-7832-01

For More Info: IBM Manuals

- IBM z Server zSeries 990 (z990) Cryptography Implementation SG24-7070-00
- z/OS V1R9.0 Cryptographic Services ICSF System Programmer's Guide SA22-7520-10
- DB2 9.1 Administration Guide SC18-9840
- Implementing PKI Services on z/OS G24-6968

For More Info:

NIST Special Publication 800-25 COMPUTERSECURITY Federal Agency Use of Public Key Technology for Digital Signatures and Authentication

(http://csrc.nist.gov/publications/nistpubs/800-25/sp800-25.pdf)

Federal Public Key Infrastructure (PKI) X.509 Certificate and CRL Extensions

Profile (National Institute of Standards and Technology 100 Bureau Dr.

Gaithersburg, MD 20899-8930)

(http://www.idmanagement.gov/fpkipa/documents/fpki_certificate_profile.pdf)