Historical cryptography

cryptography ≈ encryption

main applications: **military and diplomacy**

ancient times → **world war II**

Historical cryptography

- All “historical” cryptosystems badly broken!
- No clear understanding or science of what properties are needed.
- Honest users and attackerss are humans with limited computational capabilities.

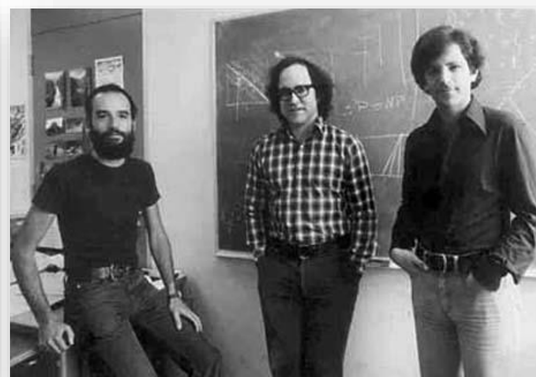
Modern cryptography

cryptography based on rigorous science/math

multiparty-computations

coin-tossing

zero-knowledge



electronic auctions

electronic voting

e-cash

private info retreival

threshold crypto

...

information theory

public-key cryptography

signature schemes

rigorous definitions

post-war

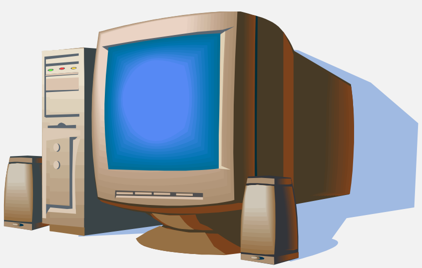
sevenites

now

What happened?

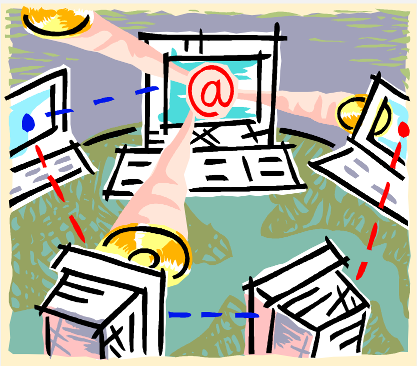
Technology

afforadable hardware



Demand

companies and individuals start to do business electronically



Theory

information theory + computational complexity

can reason about security in a **formal way**.

Modern cryptography

- Rigorous definitions of what it means to have secure encryption, signature, …
- Elegant constructions using number theory, algebra. (Still many ad-hoc constructions, we'll ignore them)
- Proofs of security
  - usually rely on simple-to-state, well-studied "hardness assumption".

Provable security – the motivation

In many areas of computer science formal proofs are **not essential**.

For example, instead of proving that an algorithm is efficient,
we can just simulate it on a "***typical*** input".

In **cryptography** we can't experimentally demonstrate security.
A notion of a "***typical*** attacker" does not make sense.
Can't run a test to check non-existence of an attack.

Need proofs!

This course is about...

- <u>Main focus:</u> how can we rigorously define security requirements, reason about them, use math to achieve them?

- <u>Cover</u>: basic cryptographic primitives: encryption, authentication, hash functions, signatures...
  - Some advanced topics, mostly towards the end.
  - Emphesize elegant ideas and constructions over ad-hoc methods and schemes used in practice.

This course is **not** about

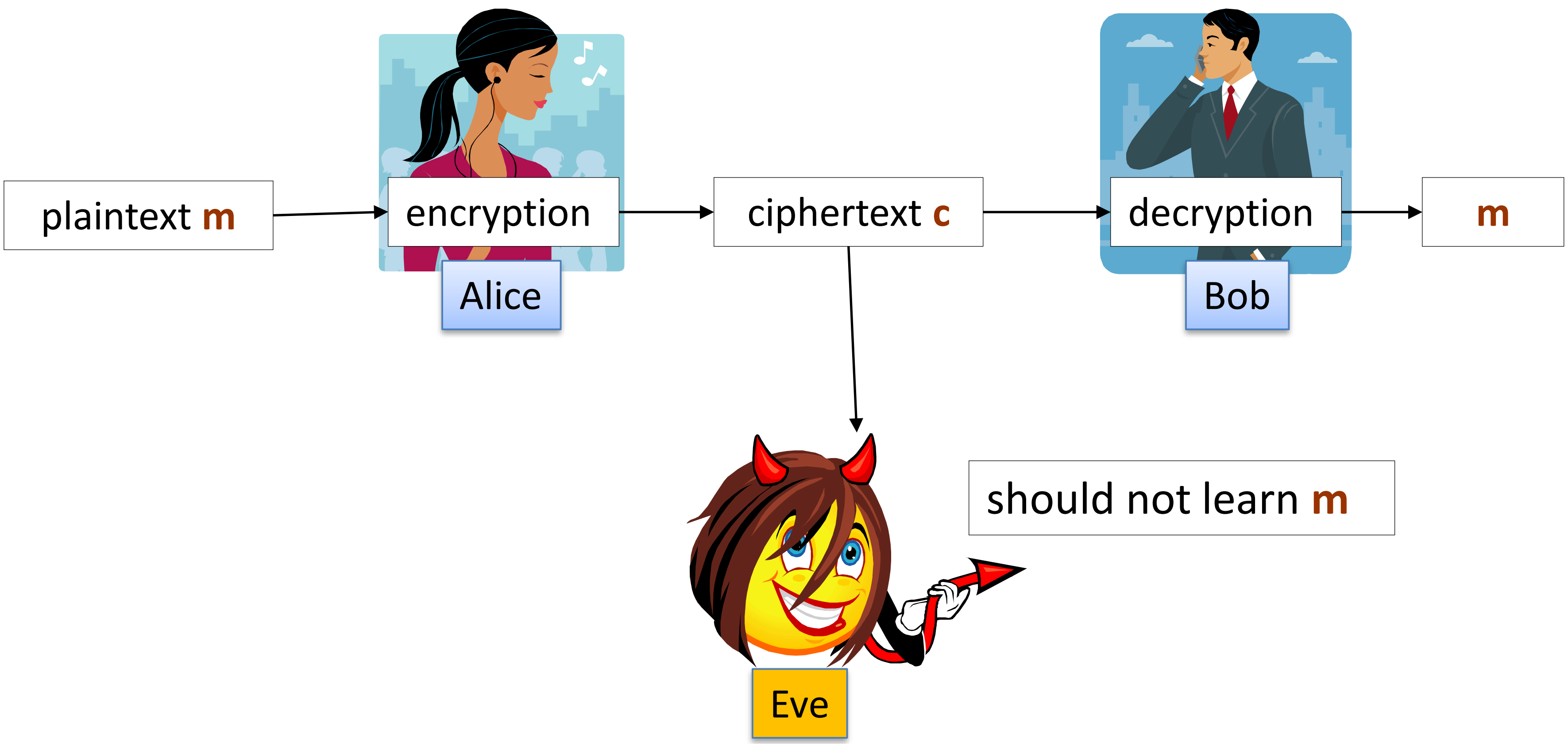
- **practical data security** (firewalls, intrusion-detection, VPNs, etc.),
- **Implementing cryptography**: many pitfalls
- **history** of cryptography,
- **number theory** and **algebra**

  (we will use them **only as tools**)
- **complexity theory.**

The Encryption Problem

Encryption Schemes
(a very general picture)

Encryption scheme = encryption & decryption procedures



Kerckhoffs' principle

Auguste Kerckhoffs (1883):

The enemy knows the system

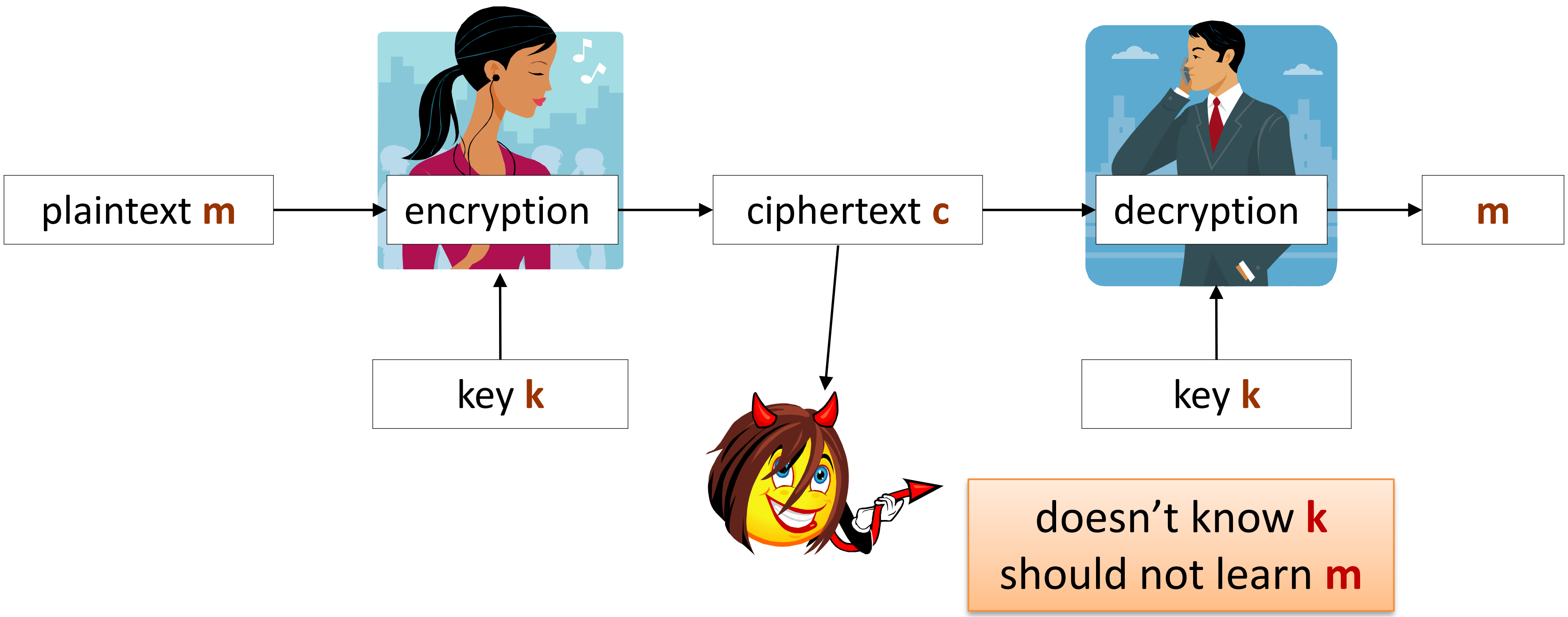
The cipher should remain secure even if **the adversary knows the specification of the cipher.**

The only thing that is **secret** is a

key **k**

that is **usually chosen uniformly at random**

A more refined picture



Kerckhoffs' principle: motivation

1. It is unrealistic to assume that the design details remain secret. Too many people need to know. Software/hardware can be **reverse-engineered!**
2. Pairwise-shared keys are easier to **protect**, **generate** and **replace**.
3. The design details can be discussed and **analyzed in public**.
4. What would it even mean formally that the specification is unknown? Does it have a **distribution**?

Not respecting this principle

=

``**security by obscurity**”.

A mathematical view

$\mathcal{K}$ – **key** space:

$\mathcal{M}$ – **plaintext** space

$\mathcal{C}$ - **ciphertext** space

An **encryption scheme** is a pair **(Enc,Dec)**, where

- **Enc :** $\mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$ is an **encryption** algorithm,
- **Dec :** $\mathcal{K} \times \mathcal{C} \rightarrow \mathcal{M}$ is an **decryption** algorithm.

We will sometimes write $\mathbf{Enc_k(m)}$ and $\mathbf{Dec_k(c)}$ instead of **Enc(k,m)** and **Dec(k,c)**.

Correctness

for every **k, m** we should have $\mathbf{Dec_k(Enc_k(m)) = m}$.

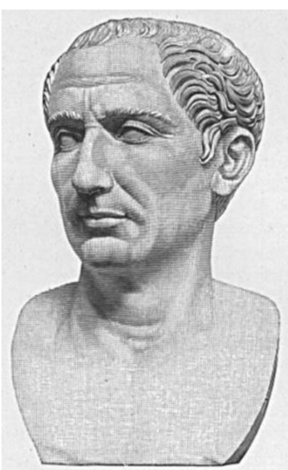
Idea 1: Shift cipher

$\mathcal{M}$ = **words over alphabet** $\{A,\ldots,Z\} \approx \{0,\ldots,25\}$

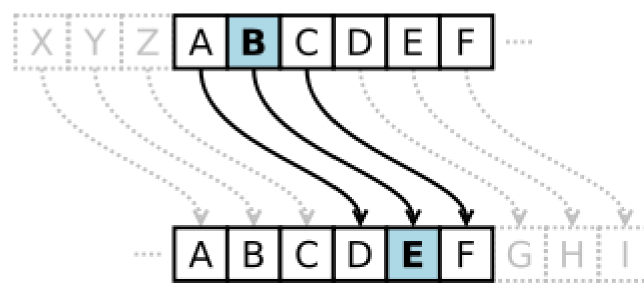
$\mathcal{K} = \{0,\ldots,25\}$

$$\mathrm{Enc}_k(m_0,\ldots,m_n) = (k+m_0 \bmod 26,\ldots,\ k+m_n \bmod 26)$$

$$\mathrm{Dec}_k(c_0,\ldots,c_n) = (k+c_0 \bmod 26,\ldots,\ k+c_n \bmod 26)$$



Cesar: $k = 3$



Security of the shift cipher

How to break the shift cipher?

Check all possible keys!

Let **c** be a ciphertext.

For every $\mathbf{k} \in \{0,\ldots,25\}$ check if $\mathbf{Dec_k(c)}$ “makes sense”.

Most probably only one such **k** exists.

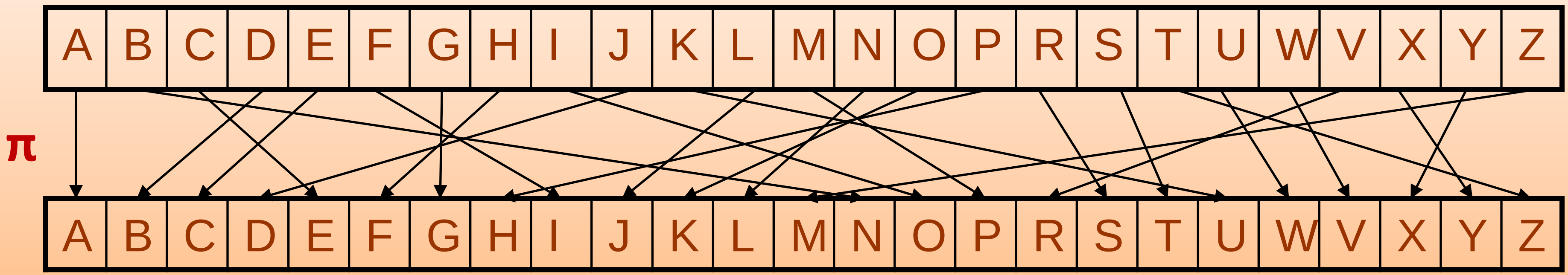
Thus $\mathbf{Dec_k(c)}$ is the message.

This is called a **brute force attack**.

Moral: the key space needs to be large!

Idea 2: Substitution cipher

$\mathcal{M}$ = words over alphabet {A,…,Z} ≈ {0,…,25}
$\mathcal{K}$ = a set of permutations of {0,…,25}



$$\mathrm{Enc}_\pi(m_0,\dots,m_n) = (\pi(m_0),\dots,\pi(m_n))$$

$$\mathrm{Dec}_\pi(c_0,\dots,c_n) = (\pi^{-1}(c_0),\dots,\pi^{-1}(c_n))$$

How to break the substitution cipher?

Use **statistical patterns** of the language.

For example: the **frequency tables**.

Texts of **50** characters can usually be broken this way.

Letter	Frequency
E	0.127
T	0.097
I	0.075
A	0.073
O	0.068
N	0.067
S	0.067
R	0.064
H	0.049
C	0.045
L	0.040
D	0.031
P	0.030
Y	0.027
U	0.024
M	0.024
F	0.021
B	0.017
G	0.016
W	0.013
V	0.008
K	0.008
X	0.005
Q	0.002
Z	0.001
J	0.001

Figure 7 - Frequency Table

Other famous "bad" ciphers

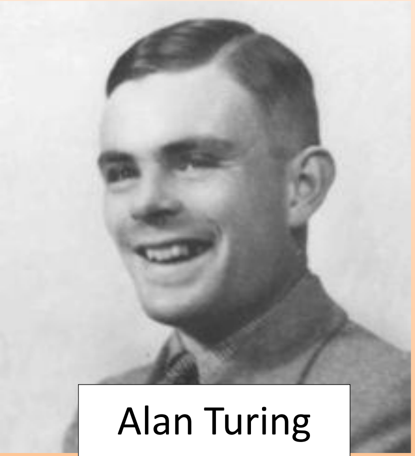
Vigenère cipher:

Blaise de Vigenère
(1523 - 1596)

Leon Battista Alberti
(1404 – 1472)

Enigma

Marian Rejewski
(1905 - 1980)



Alan Turing
(1912-1954)

Perfectly Secure Encryption

Constructions & Limitations

Defining “security of an encryption scheme” is not trivial.

consider the following experiment

(**m** – a message)

1. the key **K** is chosen uniformly at random
2. **$C := Enc_K(m)$** is given to the adversary

how to define security

?

Idea 1

(**m** – a message)

1. the key **K** is chosen uniformly at random
2. **C := Enc$_K$(m)** is given to the adversary

An idea

"The adversary should not be able to learn **K**."

A problem

the encryption scheme that "doesn't encrypt":

$$\mathbf{Enc_K(m) = m}$$

satisfies this definition!

Idea 2

($\mathbf{m}$ – a message)

1. the key $\mathbf{K}$ is chosen uniformly at random
2. $\mathbf{C := Enc_K(m)}$ is given to the adversary

An idea

"The adversary should not be able to learn **m**."

A problem

What if the adversary can compute, e.g., the first half of **m**?

m_1	…	$m_{\|m\|/2}$	?	…	?

Idea 3

(m – a message)

1. the key K is chosen uniformly at randomly
2. $C := Enc_K(m)$ is given to the adversary

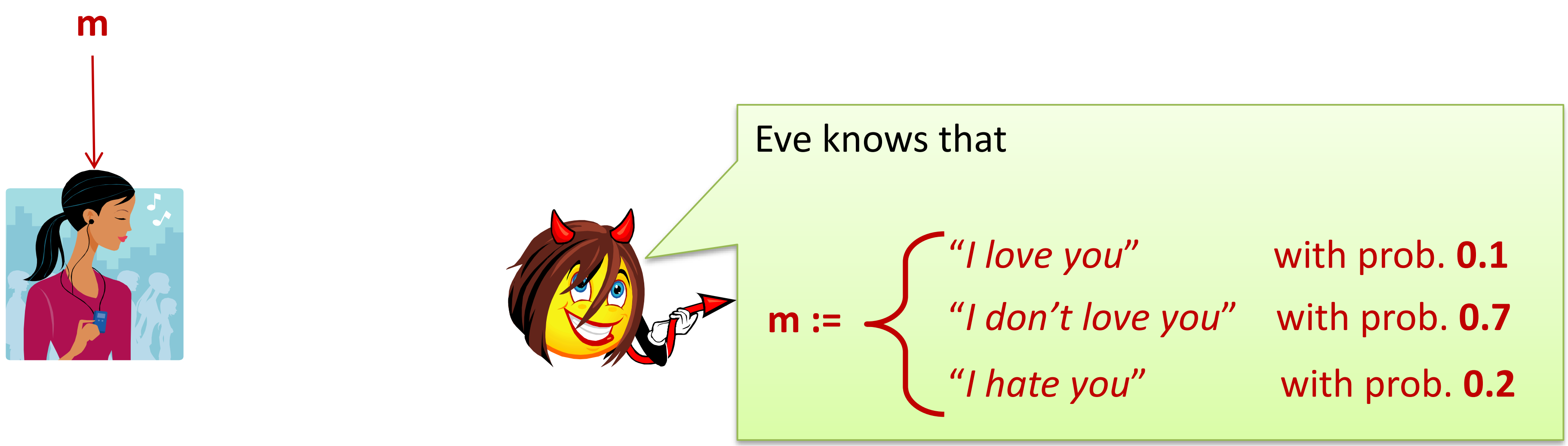
An idea

"The adversary should not learn any information about **m**."

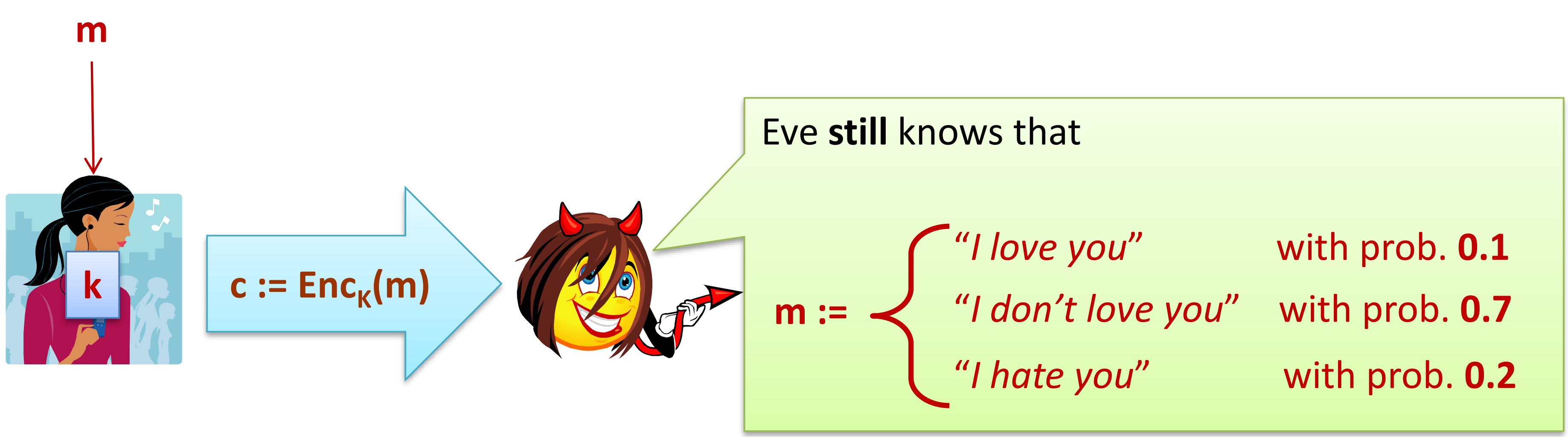
Sounds great! But what does it actually mean?
How to formalize it?

Need some probability theory.

Example



m
Eve knows that
m := “I love you” with prob. 0.1
“I don’t love you” with prob. 0.7
“I hate you” with prob. 0.2



m
k
c := Enc$_K$(m)
Eve **still** knows that
m := “I love you” with prob. 0.1
“I don’t love you” with prob. 0.7
“I hate you” with prob. 0.2

Probability Theory (review)

- Probability space:
  - Universe $\mathcal{U}$
  - Probability function: for all $\mathrm{u} \in \mathcal{U}$, assign $0 \leq \Pr[u] \leq 1$ such that $\sum_{u \in \mathcal{U}} \Pr[u] = 1$.

- Example: uniform distribution over $\mathcal{U} = \{0,1\}^2$ assigns $\Pr[00] = \Pr[01] = \Pr[10] = \Pr[11] = \frac{1}{4}$.

Probability Theory (review)

- Probability space:
  - Universe $\mathcal{U}$
  - Probability function: for all $\mathrm{u} \in \mathcal{U}$, assign $0 \leq \Pr[u] \leq 1$ such that $\sum_{u \in \mathcal{U}} \Pr[u] = 1$.
- Random variables: X, Y, Z, ...
  - Formally, functions $X : \mathcal{U} \rightarrow \mathcal{X}$, $Y: \mathcal{U} \rightarrow \mathcal{Y}$ ...
  - induce distributions $\Pr[X = x] = \sum_{\{u : X(u) = x\}} \Pr[u]$
- Example: uniform distribution over $\mathcal{U} = \{0,1\}^2$
  - X = first bit, Y = second bit, $\mathrm{Z} \coloneqq X + Y$, $W \coloneqq X \oplus Y$

Probability Theory (review)

- Probability space:
  - Universe $\mathcal{U}$
  - Probability function: for all $\mathrm{u} \in \mathcal{U}$, assign $0 \leq \Pr[u] \leq 1$ such that $\sum_{u \in \mathcal{U}} \Pr[u] = 1$.
- Random variables: X, Y, Z, ...
  - Formally, functions $X : \mathcal{U} \to \mathcal{X}$, $Y : \mathcal{U} \to \mathcal{Y}$ ...
  - induce distributions $\Pr[X = x] = \sum_{\{u : X(u) = x\}} \Pr[u]$
- Random variables X, Y are *independent* if for all x,y:

$$\Pr[X = x, Y = y] = \Pr[X = x] \cdot \Pr[Y = y]$$

Probability Theory (review)

- Example: uniform distribution over $\mathcal{U} = \{0,1\}^2$
  - X = first bit, Y = second bit, $Z \coloneqq X + Y$, $W \coloneqq X \oplus Y$

- Are X, Y independent?

- Are X, Z independent?

- Are X, W independent?

Probability Theory (review)

- For two random variables X, Y and outcomes x, y we define the conditional probability:

$$\Pr[X = x | Y = y] = \frac{\Pr[X=x, Y=y]}{\Pr[Y=y]}$$

- Interpretation: the probability that $X = x$ if we are told that $Y = y$.

- Example: uniform distribution over $\mathcal{U} = \{0,1\}^2$
  - X = first bit, Y = second bit, $Z := X + Y$, $W := X \oplus Y$
  - $\Pr[X = 1 \mid Z = 1] = ?$

Probability Theory (review)

- **Events:** An event E is a subset $\mathcal{U}$. We define $\Pr[E] = \sum_{\{u \in E\}} \Pr[u]$.

  Alternatively, can think of E as binary random var.

- Union bound: for any events E_1, E_2:

$$\Pr[E_1 \cup E_2] = \Pr[E_1] + \Pr[E_2] - \Pr[E_1 \cap E_2]$$
$$\leq \Pr[E_1] + \Pr[E_2]$$

- Example: uniform distribution over $\mathcal{U} = \{0,1\}^2$
  - Events E_1: first bit 1, E_2 : second bit 1.

Back to cryptography…

"The adversary should not learn any information about **m**."

Consider random variables:

M some random variable over $\mathcal{M}$

K uniformly random variable over $\mathcal{K}$

C = Enc(K, M) random variable over $\mathcal{C}$

"The adversary should not learn any information about **m**."

An encryption scheme is **perfectly secret** if

for every distribution of **M**

and every $\mathbf{m} \in \mathcal{M}$ and $\mathbf{c} \in \mathcal{C}$

such that

$P[C = c] > 0$

$$\Pr[\, M = m \,] = \Pr[\, M = m \mid C = c \,]$$

Equivalently:

For all **m**, **c**: **Pr[M = m] = Pr[M = m | C = c]**

$\Updownarrow$

M and **C=Enc(K,M)** are independent

$\Updownarrow$

For every **m** , **m'** , **c** we have:

Pr[Enc(K, m) = c] = Pr[Enc(K, m') = c]

A perfectly secret scheme: one-time pad

t – a parameter

$\mathcal{K} = \mathcal{M} = \{0,1\}^t$

Gilbert Vernam (1890 –1960)

component-wise **xor**

Vernam's cipher:

$$\mathrm{Enc}_k(m) = k \oplus m$$

$$\mathrm{Dec}_k(c) = k \oplus c$$

Correctness:

$$\mathrm{Dec}_k(\mathrm{Enc}_k(m)) = k \oplus (k \oplus m)$$

$$m$$

Generalized One-Time Pad

- One-time pad can be **generalized** to any finite *group*.

- **Definition:** A group **(G,+)** consists of a set **G** and an operation $+ : G \times G \rightarrow G$
  - **Associative:** $(x + y) + z = x + (y + z)$
  - **Commutative** (abelian group)**:** $x + y = y + x$
  - **Identity:** there is an element **0** s.t. $\mathbf{0} + x = x$.
  - **Inverses:** for all x, there is (- x) such that $x - x = \mathbf{0}$.

Generalized One-Time Pad

- Examples of finite groups:
  - $\mathbb{Z}_n = \{0, \ldots, n-1\}$ with addition modulo n.
    - When $n = 2$, this is bits with the xor operation!
  - ${\mathbb{Z}_n}^t$ vectors of length t, component-wise addition.
    - The zero element is $\mathbf{0} = (0, \ldots, 0)$

Generalized One-Time Pad

One time pad can be **generalized** as follows.

Let **(G,+)** be a finite abelian group.
Let $\mathcal{K} = \mathcal{M} = \mathcal{C} = \mathbf{G}$.

The following is a perfectly secret encryption scheme:

- **Enc(k, m) = m + k**
- **Dec(k, c) = c – k**

Perfect secrecy of the one-time pad

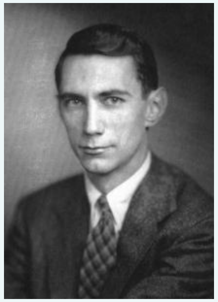
- **Theorem:** The one-time pad over a finite group **(G,+)** satisfies perfect secrecy.
- **Proof:** For any $m,\ m', c\ \in G$:

$$
\begin{aligned}
&\Pr[\ \mathrm{Enc}(K,m) = c\ ] \\
&= \Pr[K + m = c] \\
&= \Pr[K = c\ -\ m] \\
&= \frac{1}{|G|} \\
&= \Pr[\ \mathrm{Enc}(K,m') = c\ ]
\end{aligned}
$$

Why the one-time pad is not practical?

1. **The key is as long as the message.**
2. **The key cannot be reused.**
3. **Alice and Bob must share a secret key unknown to Eve.**

All three are necessary for perfect secrecy!



Theorem (Shannon 1949)

"One time-pad is optimal"

In every perfectly secret encryption scheme

$$\textbf{Enc} : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}, \quad \textbf{Dec} : \mathcal{K} \times \mathcal{C} \rightarrow \mathcal{M}$$

we have $|\mathcal{K}| \geq |\mathcal{M}|$.

Intuitive Proof:

Otherwise can do "exhaustive search". Given ciphertext c, try decrypting with every key k. Will rule-out at least 1 message.

Formal Proof:

Let M be the uniform distribution over $\mathcal{M}$ and c be some ciphertext such that $\Pr[C = c] > 0$.

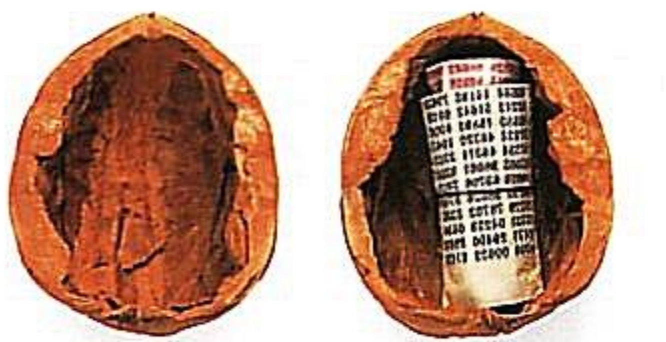
Consider the set $\mathcal{M}' = \{ \textbf{Dec}(k, c) : k \in \mathcal{K} \}$.

If $|\mathcal{K}| < |\mathcal{M}|$ then exists $m \in \mathcal{M} / \mathcal{M}'$. We have:

$$\Pr[M = m \mid C = c] = 0, \Pr[M = m] = 1/|\mathcal{M}|.$$

Practicality?

Generally, the **one-time pad** is **not very practical**, since the key has to be as long as the **total** length of the encrypted messages.



a **KGB** one-time pad hidden in a walnut shell

However, it is sometimes used because of the following advantages:

- **perfect secrecy**,
- short messages can be encrypted using **pencil and paper** .

In the 1960s the Americans and the Soviets established a hotline that was encrypted using the one-time pad.

Venona project (1946 – 1980)

Ethel and Julius Rosenberg

American **National Security Agency** decrypted **Soviet** messages that were transmitted in the 1940s.

That was possible because the Soviets reused the keys in the one-time pad scheme.

Beyond Perfect Secrecy

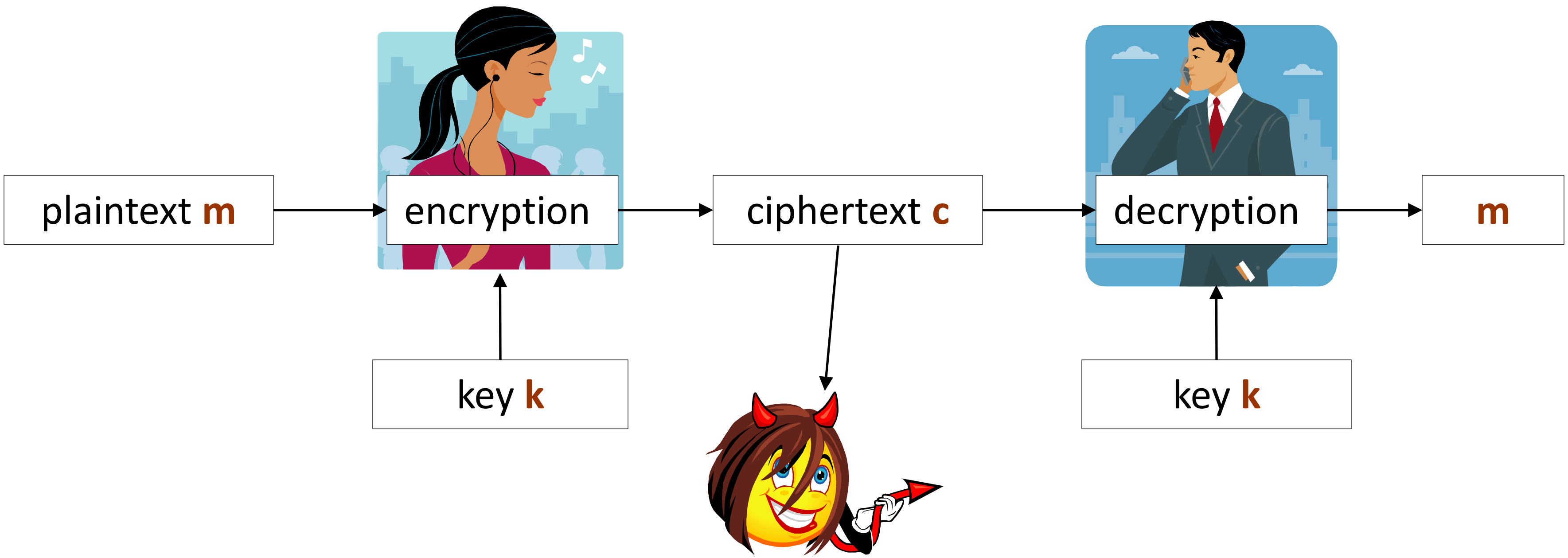
- Need to move beyond perfect secrecy to get around Shannon's result.

- Intuitively, $|\mathcal{K}| < |\mathcal{M}|$ means that exhaustive search over keys will reveal something about message. But this might not be efficient!
  - e.g., key is 128-bits, message is 10 GB.

- Will study: Secrecy against *computationally-bounded* attackers.

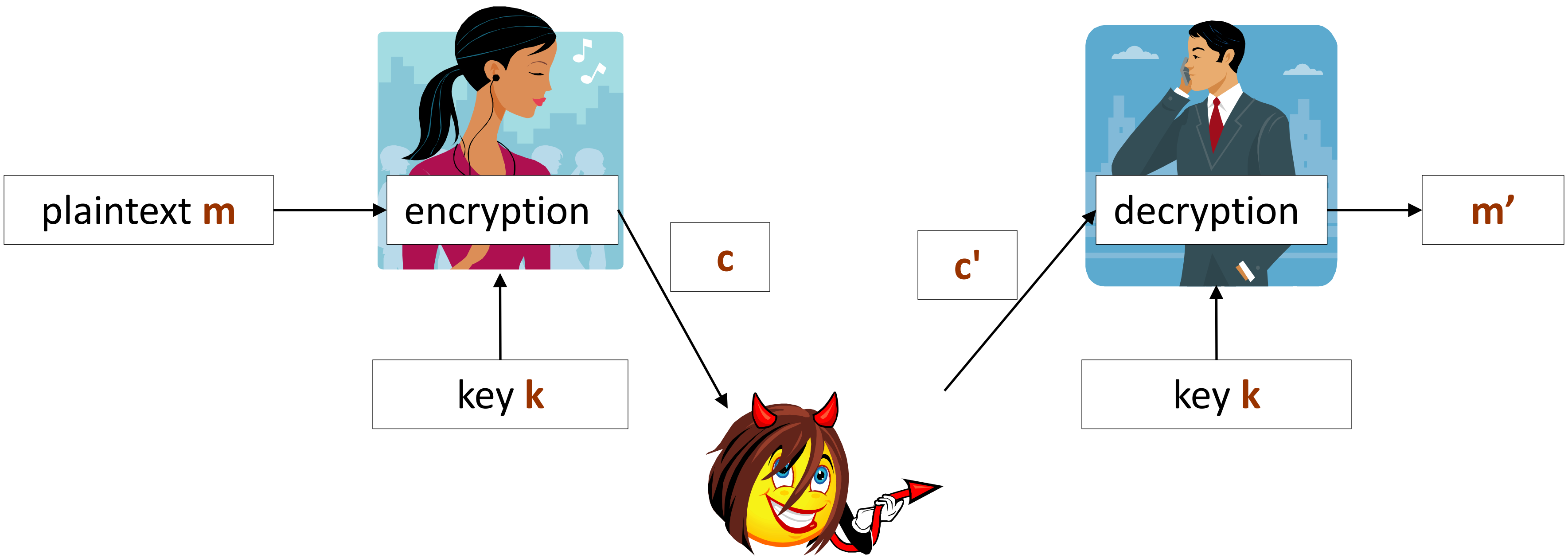
The Authentication Problem

Encryption Is Not Enough



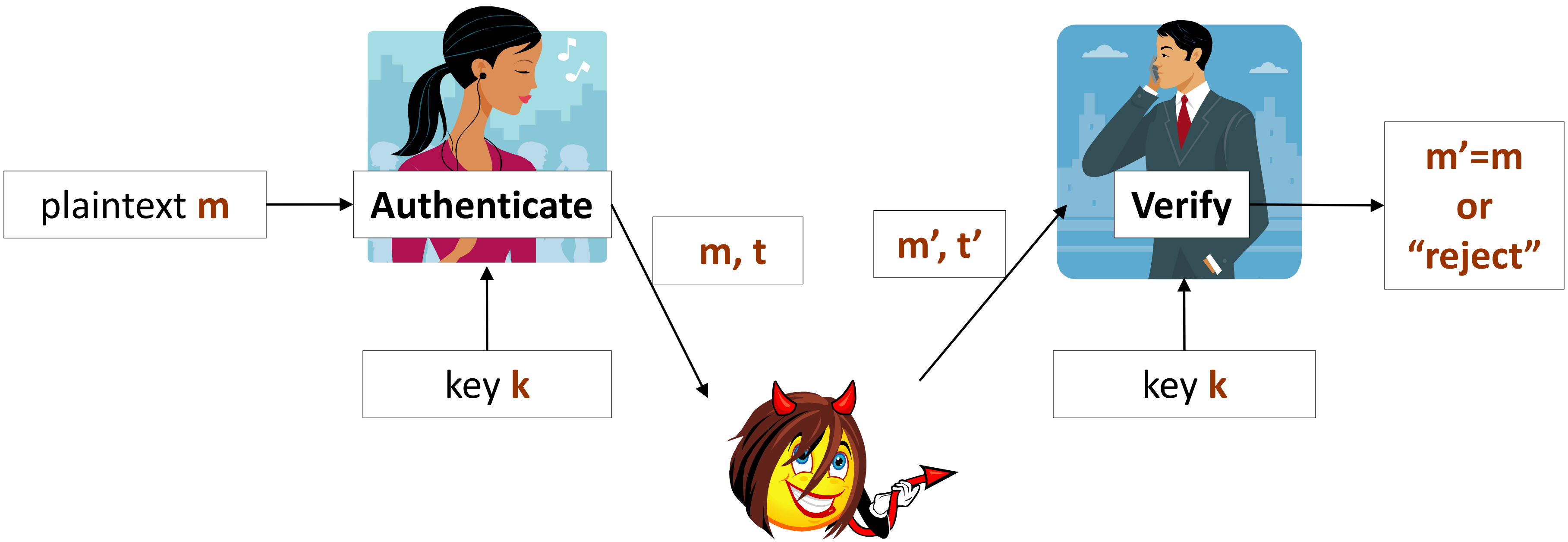
- Alice sends a 1-bit "vote" to Bob: 0 = 'no', 1 = 'yes'.
- Alice encrypts with a one-time pad: vote stays secret from Eve.

Encryption Is Not Enough



- Alice sends a 1-bit "vote" to Bob: 0 = 'no', 1 = 'yes'.
- Alice encrypts with a one-time pad: vote stays secret from Eve.
- What if Eve modifies ciphertext?
    - $c'=0$ results in random vote. $c' = c \oplus 1$, flips vote.

Authentication



Message Authentication Code (MAC)

Message space: $\mathcal{M}$, Key space: $\mathcal{K}$, Tag space: $\mathcal{T}$

- **MAC :** $\mathcal{K} \times \mathcal{M} \rightarrow \mathcal{T}$

- Usage:
  – Alice computes $t = \mathbf{MAC}(k, m)$, sends (m, t) to Bob.
  – Bob receives (m', t') and checks if $t' = \mathbf{MAC}(k, m')$.

Message Authentication Code (MAC)

Message space: $\mathcal{M}$, Key space: $\mathcal{K}$, Tag space: $\mathcal{T}$

- **MAC :** $\mathcal{K} \times \mathcal{M} \to \mathcal{T}$

- <u>Definition:</u> 1-Time Statistically Secure MAC
  - A uniformly random key k from $\mathcal{K}$ is selected.
  - Eve chooses message m and is given t = **MAC**(k, m).
  - Eve chooses (m', t') s.t. m' $\neq$ m and **wins** if t' = **MAC**(k,m').

ε-security: Pr[Eve wins] $\leq \varepsilon$

Can we make $\varepsilon = 0$?

Useful Tool: Fields

Definition: A field **(F, +, ·)** consists of a set **F** and an addition (**+**) and multiplication (·) operations.

- Operations **+,** · are **associative** and **commutative**.
- **Distributive**: $x \cdot (y + z) = x \cdot y + x \cdot z$
- **(F,+)** is a group with identity **0**.
  - For all x: $x + 0 = x$
  - For all x exists $(-x)$ such that $x - x = 0$.
- **(F*, ·)** is a group with identity **1** where **F* = F/{0}**.
  - For all x ∈**F***: $x \cdot \mathbf{1} = x$
  - For all x ∈**F*** exists (x^{-1}) such that $x \cdot x^{-1} = \mathbf{1}$.

Useful Tool: Fields

Examples of infinite fields:

- rational $\mathbb{Q}$, reals $\mathbb{R}$, complex $\mathbb{C}$.
- Not the integers!

There are finite fields.

- If p is a prime number then $\mathbb{Z}_p$ is a finite field.
- Not true when p is not a prime.

MAC Construction

Let p be a prime number.

Message/Tag space: $\mathcal{M} = \mathcal{T} = \mathbb{Z}_p$

Key space: $\mathcal{K} = \mathbb{Z}_p \times \mathbb{Z}_p$.

<u>Define:</u>

MAC$(k, m) = x \cdot m + y$ where $k = (x, y)$.

Proof of MAC Security

MAC$(k,m) = x\cdot m + y$ where $k=(x,y)$, field=$\mathbb{Z}_p$.

Theorem: Above MAC has 1-time security with $\varepsilon = \frac{1}{p}$.

Proof: Let $K = (X,Y)$ be uniformly random.

For any m any t:

$$\Pr[\mathbf{MAC}(K,m)=t] = \Pr[X\cdot m + Y = t] = \frac{1}{p}.$$

For any $m \neq m'$ any t,t':

$$\Pr[\mathbf{MAC}(K,m')=t', \mathbf{MAC}(K,m)=t]$$

$$=\Pr[X\cdot m' + Y = t',\ X\cdot m + Y = t]$$

$$=\Pr[X = x,\ Y = y] = \frac{1}{p^2} \quad \text{where } x = \frac{t-t'}{m-m'}, y = t - x\cdot m$$

Therefore: $\Pr[\mathbf{MAC}(K,m') = t' \mid \mathbf{MAC}(K,m) = t] = \frac{1}{p}$.

Practicality?

Let p be a prime number.

Message/Tag space: $\mathcal{M} = \mathcal{T} = \mathbb{Z}_p$ Key space: $\mathcal{K} = \mathbb{Z}_p \times \mathbb{Z}_p$.

MAC$(k, m) = x \cdot m + y$ where $k = (x, y)$.

Can do MUCH better!

- Construction is not very practical:
  - Key is twice as big as the message.
  - Can only use key once to authenticate single message.

Better MAC Construction

- Key space: $\mathcal{K} = \mathbb{Z}_p \times \mathbb{Z}_p$.
- Message $\mathcal{M} = \mathbb{Z}_p^d$ for any $d \geq 1$.
- Tag space: $\mathcal{T} = \mathbb{Z}_p$

For $k = (x, y)$ and $m = (m_1, \ldots, m_d)$

define **MAC**$(k, m) : \sum_{i=1}^{d} m_i x^i + y$

Proof of MAC Security

MAC$(k,m): \sum_{i=1}^{d} m_i x^i + y$ where $k = (x,y)$, field=$\mathbb{Z}_p$

Theorem: Above MAC has 1-time security with $\varepsilon = \frac{d}{p}$.

Proof: Let $K = (X,Y)$ be uniformly random.

For any m any t:

$$\Pr[\mathbf{MAC}(K,m) = t] = \Pr[\textstyle\sum_{i=1}^{d} m_i X^i + Y = t] = \frac{1}{p}.$$

For any $m \neq m'$ any t, t':

$$\Pr[\mathbf{MAC}(K,m') = t', \mathbf{MAC}(K,m) = t] \leq \frac{d}{p^2}$$

Therefore: $\Pr[\mathbf{MAC}(K,m') = t' \mid \mathbf{MAC}(K,m) = t] \leq \frac{d}{p}$.

Proof of MAC Security

MAC$(k, m) : \sum_{i=1}^{d} m_i x^i + y$ where $k = (x, y)$, field=$\mathbb{Z}_p$

<u>**Theorem:**</u> Above MAC has 1-time security with $\varepsilon = \frac{d}{p}$.

Example:

- Message size = 2^{33} bits (4 GB).
- Set $p \in [2^{128}, 2^{129}]$ just 129 bit description!
- Set $d = 2^{26}$. Think of message as d values in $\mathbb{Z}_p$.
  - $2^{26} 128 = 2^{33}$.
- Get security: $\varepsilon \leq 2^{-102}$ and key size 258 bits!

Practicality?

- Construction is *still* not very practical: can only use key once to authenticate single message.

- Unfortunately, cannot do much better if we want statistical security.

- **Theorem:** To authenticate q messages with security $\varepsilon = 2^{-r}$ need key of size $(q+1)r$.
  - Proof omitted.

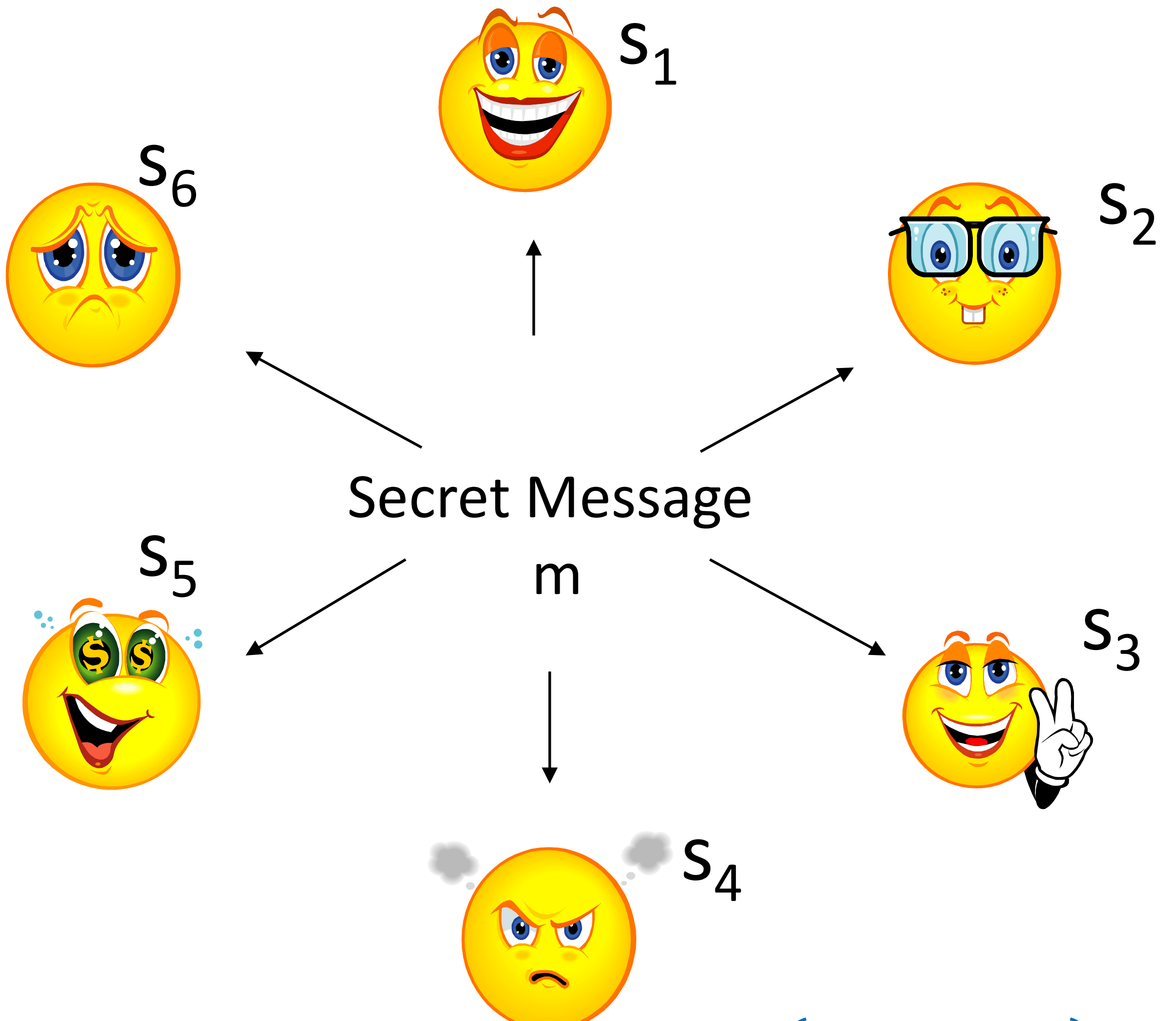
Combining Encryption & Authentication

- Can Encrypt then Authenticate ciphertext

Send: $c = \mathbf{Enc}(k_1, m)\ ,\ t = \mathbf{MAC}(k_2, c)$

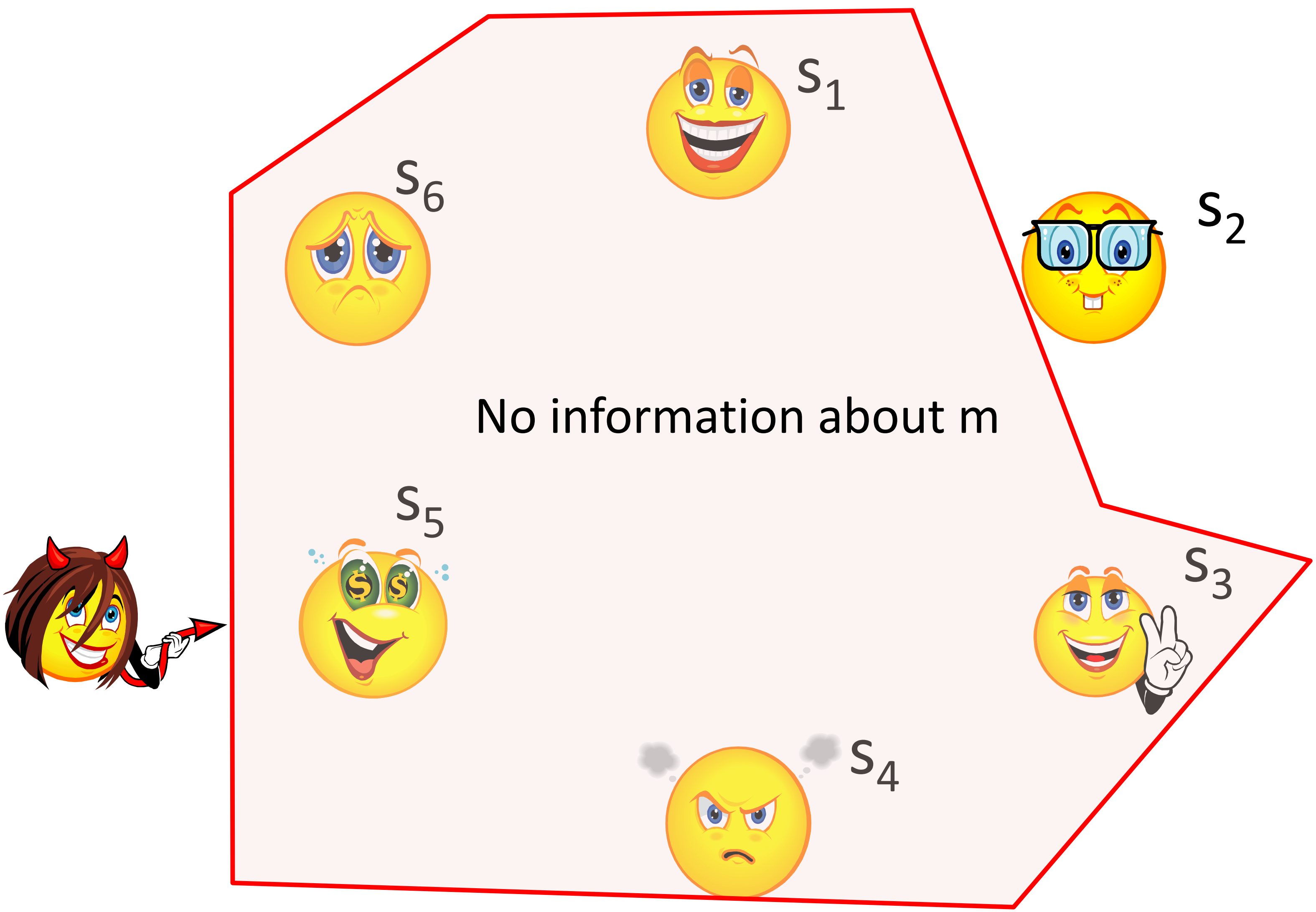
Secret Sharing

Secret Sharing

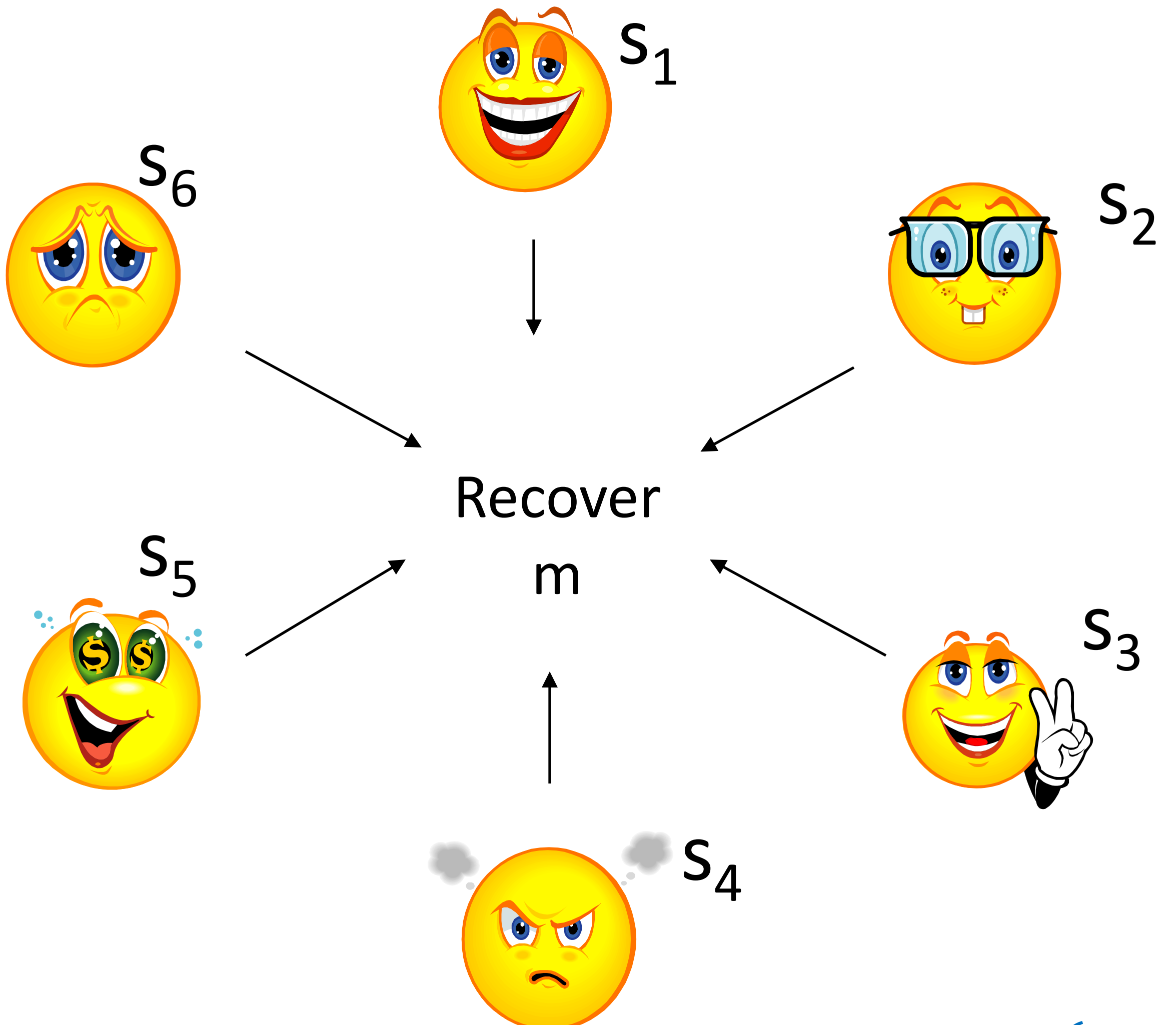


$(s_1, \dots, s_n) = \textbf{Share}(m)$

Secret Sharing



Secret Sharing



$$m = \mathbf{Rec}(s_1, \ldots, s_n)$$

Secret Sharing : Definition

Message space $\mathcal{M}$, Share space $\mathcal{S}$
Number of parties: n

Share : $\mathcal{M} \rightarrow \mathcal{S}^n$ randomized algorithm
Rec : $\mathcal{S}^n \rightarrow \mathcal{M}$

- **Correctness**: Pr[**Rec**(**Share**(m)) = m] =1
- **Perfect Security:** for all message distributions M and all sets $A \subseteq \{1, \ldots, n\}$ of size $|A| = n - 1$:
  - Let $(S_1, \ldots, S_n) =$ **Share**(M) and $S_A := \{ S_i : i \in A\}$.
  - Then the distributions of S_A and M are independent.

Secret Sharing : Construction

Message space $\mathcal{M} = \mathbb{Z}_q$,   Share space $\mathcal{S} = \mathbb{Z}_q$

Number of parties: n

Any finite group

Share(m) :

– Choose $s_1, \ldots, s_{n-1}$ uniformly at random
– Set $s_n := m - (s_1 + \cdots + s_{n-1})$

Rec$(s_1, \ldots, s_n) = s_1 + \cdots + s_n$

Theorem: Above scheme has perfect secrecy

Proof: For any dist. M, any set $A = \{1, \ldots, \mathrm{n}\}/\{\mathrm{i}\}$ and any value s_A, m, we have $\Pr[S_A = s_A \mid M = m] = \frac{1}{q^{n-1}}$.* Probability is same for all m means S_A and M are independent.

* For a fixed m, each choice of s_A corresponds to unique $s_1, \ldots, s_{n-1}$.

Threshold Secret Sharing

- Still have n parties with one share per party, but now also threshold t:
  - **Correctness**: Any $t + 1$ can recover the message.
  - **Security**: Any t don't learn anything message.

- Previous case corresponds to $t = n - 1$. Can we generalize to any t?

Threshold Secret Sharing

Construction (Shamir Secret Sharing)

- Number of parties n, Threshold $t < n$.
- Message $\mathcal{M} = \mathbb{Z}_q$, Shares $\mathcal{S} = \mathbb{Z}_q$: $q > n$ prime.

Any finite field

- **Share**(m) :
  - Choose t random "coefficients" $c_1, \ldots, c_t$ and set $c_0 := m$.
  - Define polynomial $p(x) = \sum_{j=0}^{t} c_j x^j$
  - Output $s_i = p(i)$.
- **Recover**$(\{ (i, s_i) \})$: Lagrange Interpolation.

Lagrange Interpolation

Let $z_0, \dots, z_t$ be any distinct field elements in $\mathbb{Z}_q$.

Theorem: There is an (efficiently computable) bijection between

- Coefficients: $(c_0, \dots, c_t)$ giving poly $p(x) = \sum_{j=0}^{t} c_j x^j$
- Evaluations: $s_0 = p(z_0), \dots, s_t = p(z_t)$.

Proof:

- Coefficients → evaluations: easy – evaluate!
- Evaluations → coefficients:
  - Let $p_i(x) := \prod_{j \neq i} \frac{x - z_j}{z_i - z_j}$. Then $p_i(z_i) = 1, p_i(z_j) = 0$ for $j \neq i$.
  - Let $p(x) := \sum_{i=0}^{t} s_i \cdot p_i(x)$. Then $p(z_i) = s_i$.

Threshold Secret Sharing

Construction (Shamir Secret Sharing)

- **Share**(m) :
  - Choose t random “coefficients” $c_1, \dots, c_t$ and set $c_0 \coloneqq m$.
  - Define polynomial $p(x) = \sum_{j=0}^{t} c_j x^j$
  - Output $s_i = p(i)$.
- **Recover**$(\ \{\ (i, s_i)\ \}\ )$: Lagrange Interpolation.

Theorem: Shamir Secret Sharing has perfect secrecy.

Proof: For any message m, any t distinct points $z_1, \dots, z_t \subseteq \mathbb{Z}_q/\{0\}$ and values $s_1, \dots, s_t$ we have

$$\Pr[p(z_1) = s_1, \dots, p(z_t) = s_t \mid M = m] = \frac{1}{q^t}$$

Since, once we fix $p(0) = c_0 = m$, each choice of $s_1, \dots, s_t$ corresponds to unique choice of $c_1, \dots, c_t$.

Summary

- Saw:
  - “perfectly secure” encryption, secret sharing
  - “statistically secure” message authentication
- No restrictions on attacker computational power

- Big limitations:
  - One-time use per key.
  - For encryption, $|\text{message}| < |\text{key}|$

Some of the slides and slide contents are taken from
http://www.crypto.edu.pl/Dziembowski/teaching
and fall under the following: