

**ISPEC 2006**  
**11 April 2006**

**Hangzhou, China**  
**Zhejiang Xizi Hotel**

# **Provable Security—Myth or Reality?**

**James L. Massey**

Prof.-em. ETH Zürich  
Trondhjemsgade 3, 2TH  
DK-2100 Copenhagen East

**[JamesMassey@compuserve.com](mailto:JamesMassey@compuserve.com)**

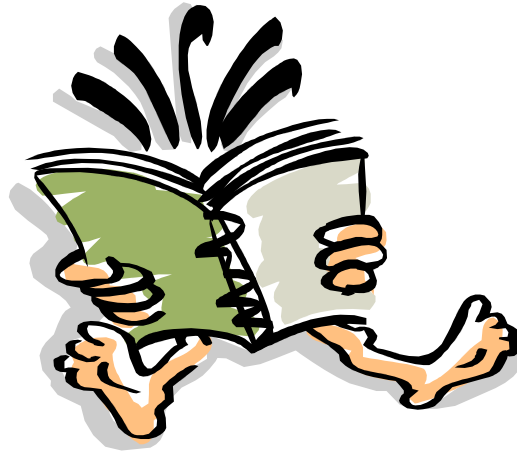
# The goals of cryptography

Secrecy

Authenticity

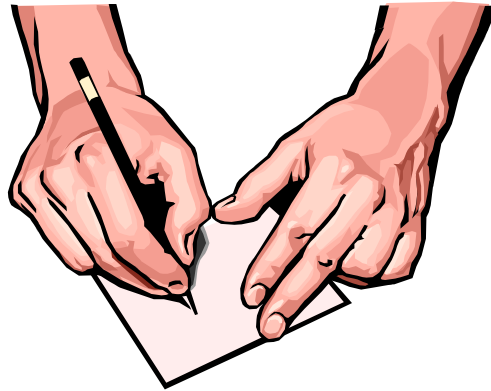
**Xuejia Lai** has given a useful **razor** for deciding whether something is a matter of secrecy or a matter of authenticity.

**Secrecy** - concerned with who has **access** to (or can **read**) a legitimate message.



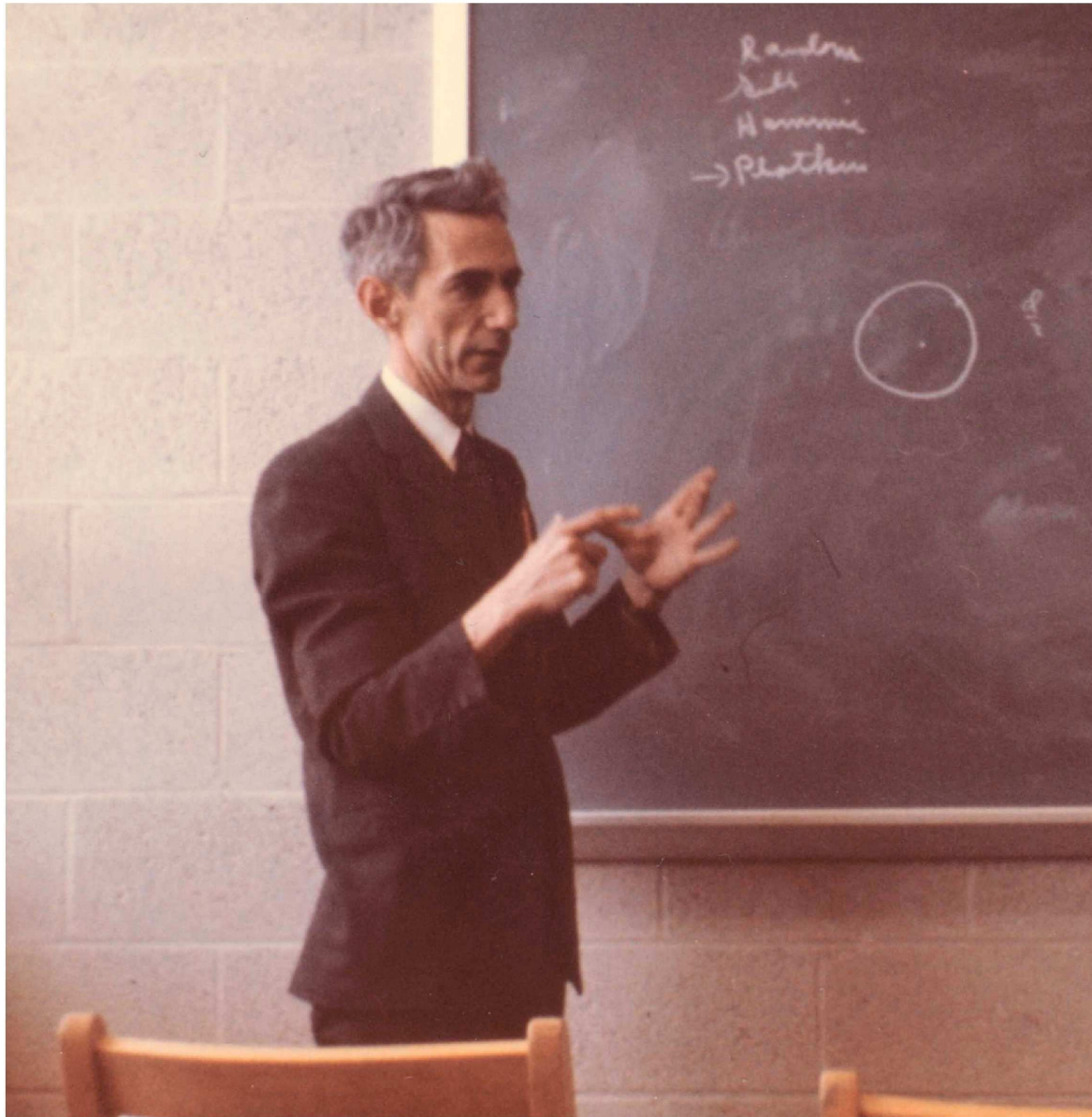
Secrecy deals with safeguarding the future by ensuring that **only authorized recipients will be able to gain access** to (or read) a legitimate message.

**Authenticity** - concerned with who can **create** (or **write**) a legitimate message.



Authenticity deals with protecting the past by

- ensuring that the creator (or author) was **entitled to create** (or write) the message
- ensuring that the **contents** of the message **have not been altered**



Claude Elwood Shannon (1916-2001)  
(photograph 17 April 1961 by Göran Einarsson)

**“As a first step in the mathematical analysis of cryptography, it is necessary to idealize the situation suitably, and to **define in a mathematically acceptable way** what we shall mean by a [cryptographic] ~~secrecy~~ system.”**

C. E. Shannon, “The communication theory of secrecy systems”, *B.S.T.J.*, 1949.

# Security

applies to both types of protection that we may be interested in, both **secrecy** and **authenticity**.



Shannon distinguished between two types of security:

- Unconditional\* (or **theoretical** as Shannon called it) - means security against an enemy who has unlimited time and computational resources.
- Computational (or **practical** as Shannon called it) - means security against an enemy who has a specified limited amount of time and computational resources.

\*Today, unconditional security is often called information-theoretical security.

We see that we must consider two answers to the question:

**What does provable security mean?**

**Provable unconditional security**: This means that the cryptographic system can be rigorously proved to provide the **claimed protection** against an **enemy who has unlimited time and computational resources**.

**Provable computational security**: This means that the cryptographic system can be rigorously proved to provide the **claimed protection** against an **enemy who has a specified limited amount of time and computational resources**.

What does claimed protection mean?

This means a careful and complete description of the operational scenario, specifying exactly

- **what the cryptographic system is intended to prevent the attacker from doing**

together with

- **what is known by the attacker**
- **what is known by the users of the system**
- **any physical assumptions** governing what the attacker and/or users can do.

The thesis of my talk today is that

**provable unconditional security is a reality,**

but that

**provable computational security is a myth.**

In 1949, Shannon proved the existence of secret-key secrecy systems that are **unconditionally secure** against a ciphertext-only attack.

(**Claimed security**) Shannon claimed that these systems provide "perfect secrecy" in the sense that the ciphertext is independent of the plaintext (and hence **the attacker in a ciphertext-only attack cannot do better than guessing the plaintext without observing the ciphertext**).

The next slide shows Shannon's "careful and complete description of the operational scenario" for his proof of unconditional security.

(Shannon, 1949)

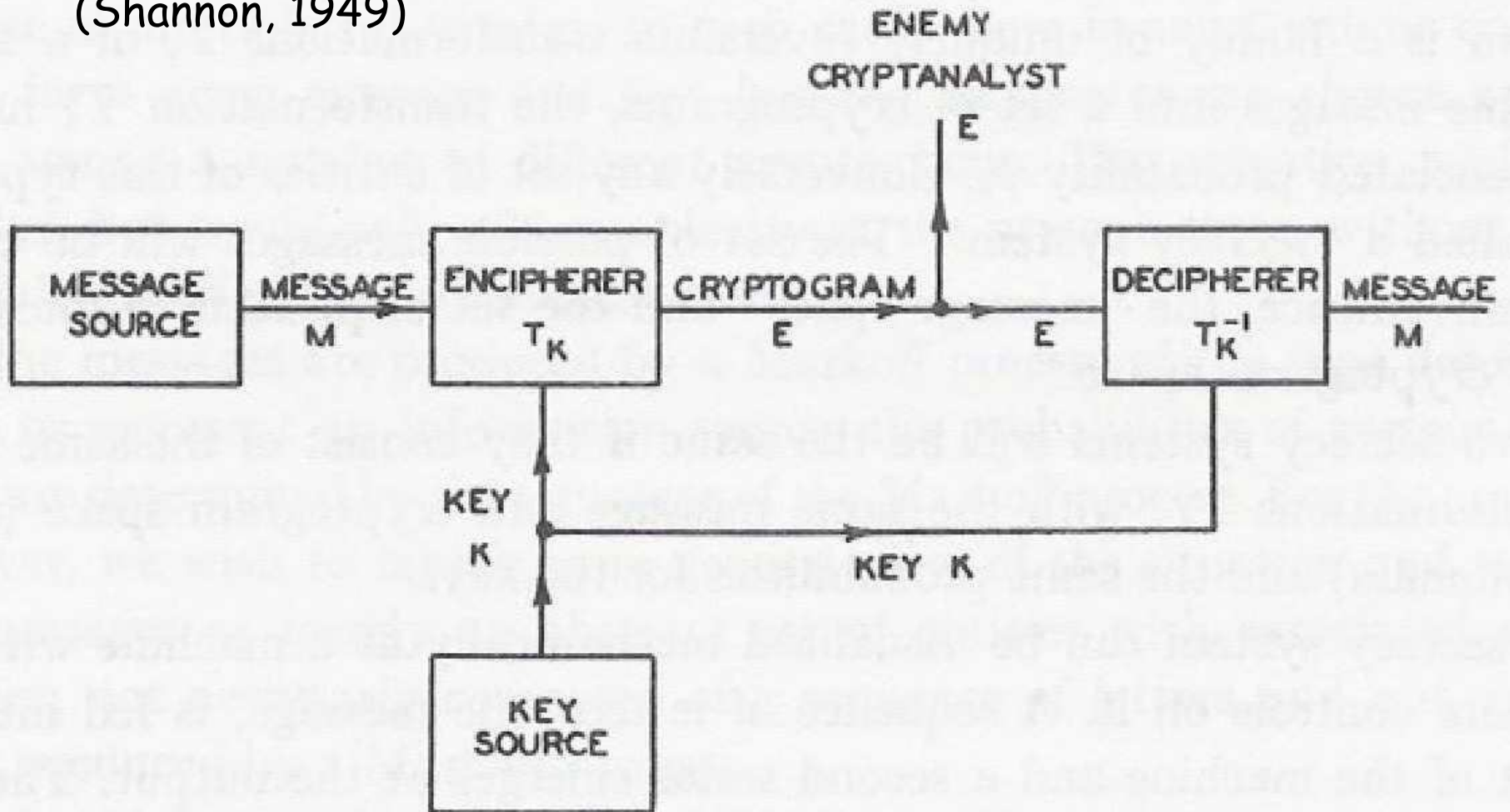


Fig. 1—Schematic of a ~~general secrecy system.~~

ciphertext-only attack on a secret-key cipher

Shannon makes the **physical assumption** that sources may be described as devices whose outputs are random processes.

If you do not accept the reality of random processes, then Shannon's proof will not be convincing!

(The output of the MESSAGE SOURCE and the KEY SOURCE in Shannon's Fig. 1 are independent random processes.)

The Binary Symmetric Source (BSS) is a device whose output is a binary coin-tossing sequence, i.e., a "completely random" binary sequence.

The Binary Symmetric Source (BSS) of information theory can be realized by a monkey with a fair binary coin (0 on one side and 1 on the other).



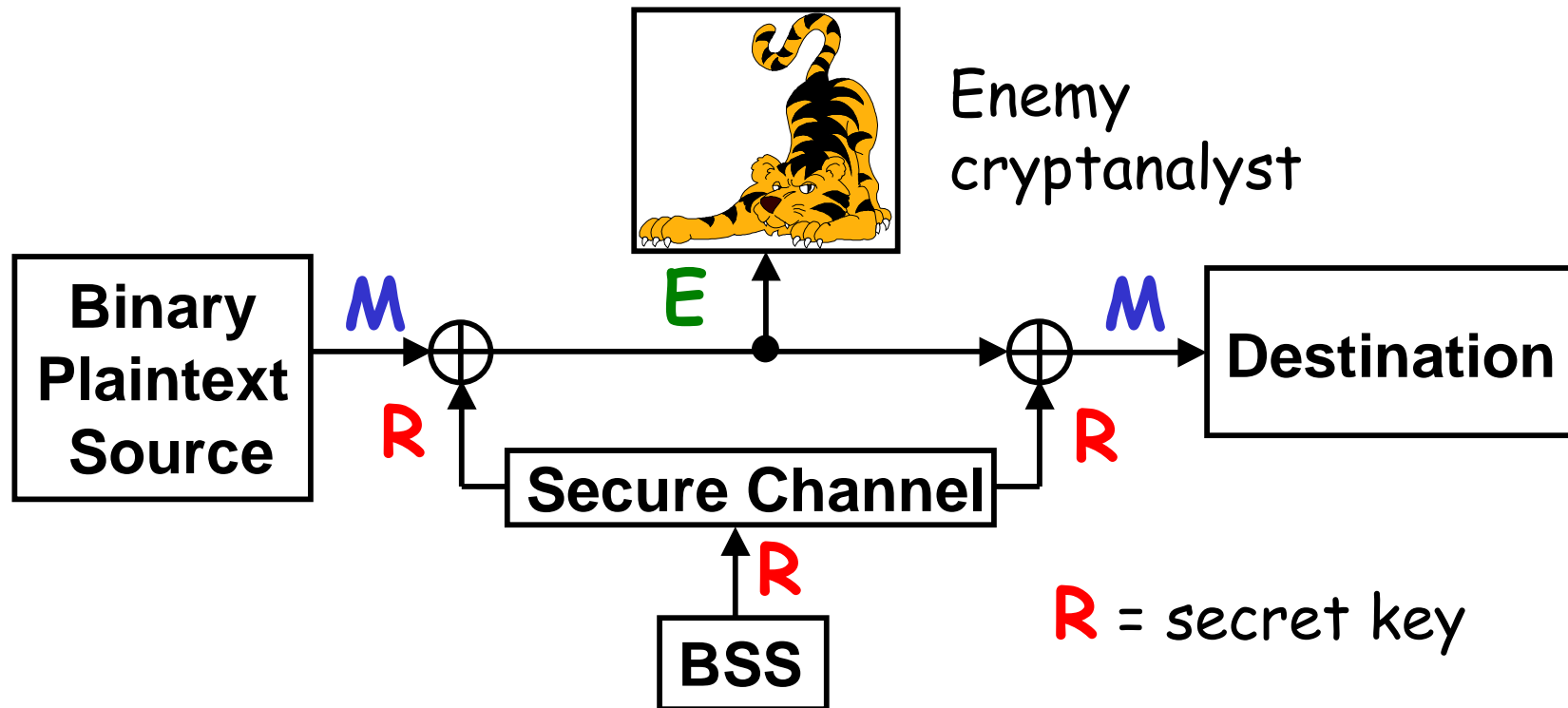
## Cryptographic property of the BSS:

The modulo-two sum of a BSS output and an arbitrary random sequence is another BSS output that is INDEPENDENT of the arbitrary random sequence.

### Example:

<b>BSS output:</b>	0	1	0	0	1	0	1	0	1	1	1	0	1	...
<b>Arb. Ran. Seq:</b>	1	1	1	1	1	1	1	1	1	1	1	1	1	...
<b>Modulo-2 sum:</b>	1	0	1	1	0	1	0	1	0	0	0	1	0	...

Vernam's 1926 cipher provides perfect secrecy against a ciphertext-only attack!



The **cryptogram  $E$**  that the enemy cryptanalyst sees is independent of the **plaintext message  $M$** . This simple **proof of perfect secrecy** for Vernam's 1926 cipher was first given by Shannon in **1949!**

Vernam's cipher needs as many binary digits of secret key as there are bits of plaintext to be encrypted. Does an unbreakable cipher (i.e., a cipher giving perfect secrecy) really need this huge amount of secret key?

**Yes!** Shannon proved in 1940 that:

For perfect secrecy, the number of different possible keys must be AT LEAST AS GREAT as the number of different possible plaintexts.

## Shannon's 1949 Proof of the Lower Bound on Key Length:

("possible" means "having non-zero probability")

Proof:

- For any fixed possible key **k**, the number of different possible ciphertexts **e** equals the number of different possible plaintexts **m**.
- Perfect secrecy  $\Rightarrow$  for all possible **e** and any fixed possible **m**,

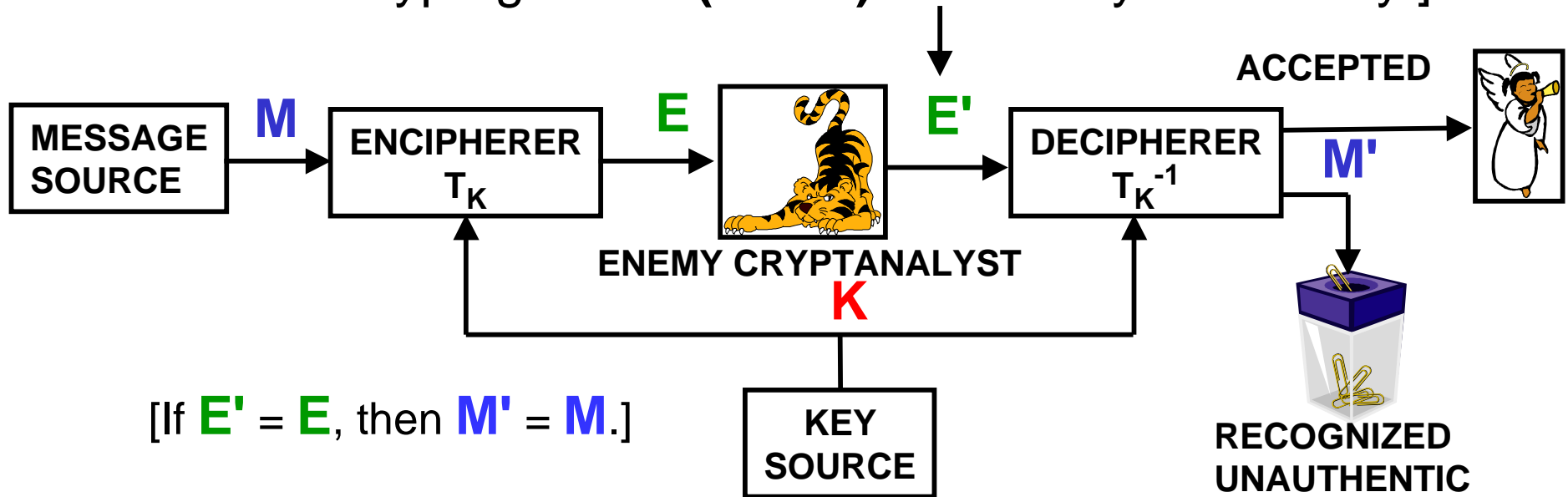
$$P(E=e|M=m) = P(E=e) \neq 0$$

- $\Rightarrow$  For a fixed possible **m**, the number of different possible ciphertexts **e** must equal at least the number of different possible plaintexts **m**.
- But all possible **keys** from a fixed **m** to different **e**'s **must be different**.

In **1984**, Simmons proved the existence of secret-key authenticity systems that are **unconditionally secure** against an optimum **substitution attack**.

G. J. Simmons, "Authentication Theory/Coding Theory," pp. 411-431 in *Advances in Cryptology - CRYPTO '84* (Eds. G. R. Blakey and D. Chaum), Lecture Notes in Computer Science No. 196. Heidelberg and New York: Springer, 1985.

[ $E'$  can be the legitimate cryptogram  $E$  or a phony cryptogram  $E'$  ( $E' \neq E$ ) created by the enemy.]



[ $E'$  is ACCEPTED if and only if it is a valid cryptogram for the key  $K$ .]

## Simmons' 1984 Model of a **Substitution Attack** on an **Authenticity System**

(but Simmons did **not** himself draw such a picture!)

In an substitution attack, the attacker forms a phoney cryptogram  $E'$  after seeing one legitimate cryptogram  $E$  and wins if  $E' \neq E$  and his phoney cryptogram  $E'$  is accepted.

$P_S$  = Probability of successful substitution when the attacker uses an optimum attack.

Simmons' 1984 bound on the probability of successful substitution:

$$P_S \geq 2^{-I(E;K)}$$

where  $I(E;K) = H(K) - H(K|E)$  is the mutual information between  $E$  and  $K$ .

The only way to get unconditionally secure authenticity is to allow the cryptogram to give away information about the key!

Here is an example (not appearing in Simmons' paper) to show that Simmons' lower bound on  $P_S$  can be achieved.

$$\mathbf{M} = [M_1, M_2, \dots, M_L] \quad \text{L-bit message}$$

$$\mathbf{K} = [K_0, K_1, K_2, \dots, K_L] \quad \text{(L+1)n-bit key}$$

The n-bit subkeys  $K_0, K_1, K_2, \dots, K_L$  are independent random sequences from a BSS. The cryptogram is

$$\mathbf{E} = [M_1, M_2, \dots, M_L, S]$$

where the "signature"  $S = K_0 + M_1K_1 + M_2K_2 + \dots + M_LK_L$  and where the additions are bit-by-bit modulo-two.

After observing  $E$ , the attacker knows  $M$  and can choose any  $M' \neq M$  that he wishes. To win, he must be able to form  $S' = K_0 + M'_1 K_1 + M'_2 K_2 + \dots + M'_L K_L$ , or equivalently (since he knows  $S$ ) to form

$$S - S' = (M_1 - M'_1) K_1 + (M_2 - M'_2) K_2 + \dots + (M_L - M'_L) K_L.$$

But at least one of the binary coefficients  $(M_j - M'_j)$  must be a 1 and hence, by the cryptographic property of the BSS, the  $n$ -bit sequence  $S - S'$  is also a BSS sequence so the probability that the attacker can guess it correctly is

$$P_S = 2^{-n}.$$

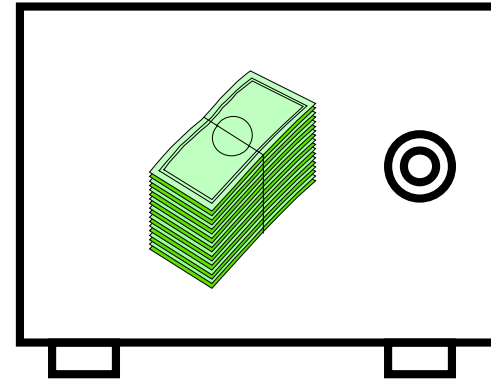
It is easy to check that  $I(E; K) = n$  bits so that **Simmons' lower bound on  $P_S$  holds with equality!**

In 1979, Shamir proved the existence of secret-key secret-sharing schemes that are **unconditionally secure** against an **attacker who can acquire only a specified number of shares of the secret.**

A. Shamir, "How to share a secret", *Communications of the ACM*, vol. 22, pp. 612-613, November 1979.

## Secret Sharing

The "classical" way that two crooks (or two bank vice presidents), who do not trust one another, can share a secret.



The **secret**:

1 0 0 1 0 1 1 0 0 1

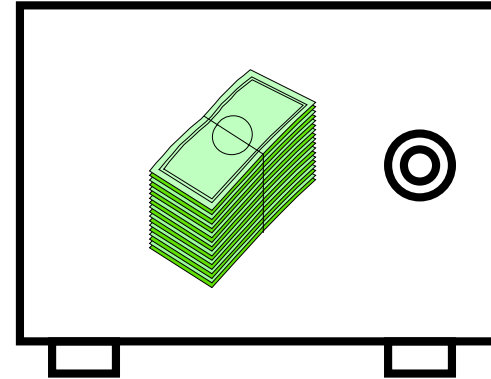
The  
**shares**

1 0 0 1 0

1 1 0 0 1

The **secret "leaks out"**—one share is not worthless!

## No-Leak Secret Sharing



The **secret**

1 0 0 1 0 1 1 0 0 1

**Share 1:**

**BSS output**

0 0 1 1 0 1 0 1 1 1

**Share 2:**

**secret  $\oplus$  BSS output**

1 0 1 0 0 0 1 1 1 0

No leakage!

(Share 2 is a Vernam encryption of the secret.)

What did Shannon have to say  
about **computational security**  
of a secrecy system?

**“The problem of good cipher design is essentially one of finding difficult problems,** subject to certain other conditions.

. . .

“How can we ever be sure that a system which is not ideal and therefore has a unique solution for sufficiently large  $N$  will require **a large amount of work to break with every method of analysis?** ... **We may construct our cipher in such a way that breaking it is equivalent to (or requires at some point in the process) the solution of some problem known to be laborious.”**

(Shannon, 1949)

A sobering thought:

Shannon was unable to prove anything interesting about computational security!

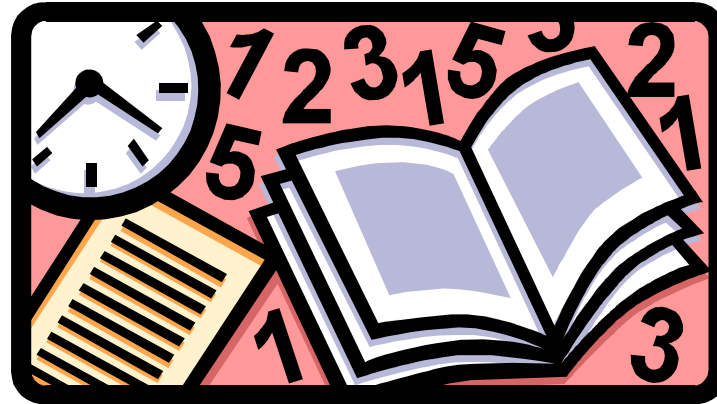
In my opinion, provable computational security is a myth! Not only do we have no proofs of computational security today, but we are so far from such proofs that it seems unlikely that we will have any in the foreseeable future—if ever!

**Where can we look for help** in determining the difficulty of a problem as part of a proof of computational security?



- **Number theory?**
- Theoretical computer science, i.e., **computational-complexity theory?**
- **Gate complexity theory?**

# Number theory?



## Forget it!

Nobody has ever proved an interesting lower bound on the complexity of doing any particular apparently difficult task (like taking the discrete logarithm in a finite field or on an elliptic curve, or factoring the product of two large distinct primes) in number theory. Nor is anyone likely to do this in the next millenium.

# Theoretical computer science?

In computational complexity theory, *Problems* (or *functions*) must have countably infinitely many instances of increasing size, each of which is a **problem** (or a **function**).

[In simpler words, a *problem* (or *function*) is a countably infinite family of **problems** (or **functions**)] whose sizes grow without bound.]

Example (Jevon's **problem**, 1873):

$m = 8\,616\,460\,799$  is the product of two distinct primes, what are they?

Jevon stated that "I think it is unlikely that anyone will ever know; for they are two large prime numbers."

$$\text{N. B.: } 8\,616\,460\,799 = 96\,079 * 89\,681$$

Example: The **problem**: Given the product  $m$  of two distinct primes  $p_1$  and  $p_2$ , find these primes.

(Jevon's **problem** is an instance of the above **problem**.)

Breaking a cryptographic system (say, the AES) is a **problem**, not a **problem**.

My opinion: **Computational complexity theory is of little or no use** in determining the computational security of cryptographic systems.

# Gate complexity of functions?

(N.B. Shannon liked to work with gate complexity!)

$P_n$  = set of permutations on  $\{0, 1\}^n$  (i.e., the set of invertible **functions** from  $n$  bits to  $n$  bits).

The **gate complexity** of a **function** in  $P_n$  is the smallest number of gates (a gate is defined as a boolean function of two variables) in an acyclic gate network that computes this **function**.

Can we find good “**one-way**” **functions** in  $P_n$ , i.e., invertible functions with **great computational asymmetry**? This would seem to be the first step on the way to a theory of cryptography for computational security.

Alain Hiltgen holds the world record for **computational asymmetry** for constructive **functions** in  $P_n$ . He can, for every  $n$ , construct a function whose inverse requires **twice as many gates** as the function itself!

In 1996 I was able to prove (with an assist from Eli Biham) the following:

**Proposition:** For all  $n \geq 6$ , **virtually all functions** in  $P_n$  have gate complexity **that differs by a factor of less than 2.5** from the gate complexity of their **inverse function**.

This is in stark contrast to Shannon's channel coding theorem in which he showed that **virtually all codes are good**.

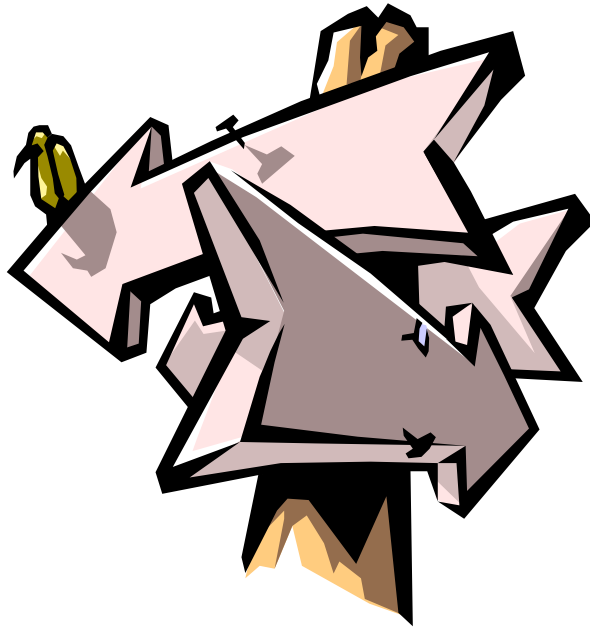
You can find a proof of the previous proposition as well as more information on gate complexity by going to <http://www.iacr.org> and following the links to the IACR Distinguished Lectures and then to my 1996 lecture.

## Is provable computational security possible?

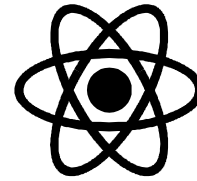
**No**, in my opinion, not with the methods of number theory and theoretical computer science.

**Maybe** someday with the methods of gate complexity but we do not even know today whether genuine one-way functions, as measured by gate complexity, exist.

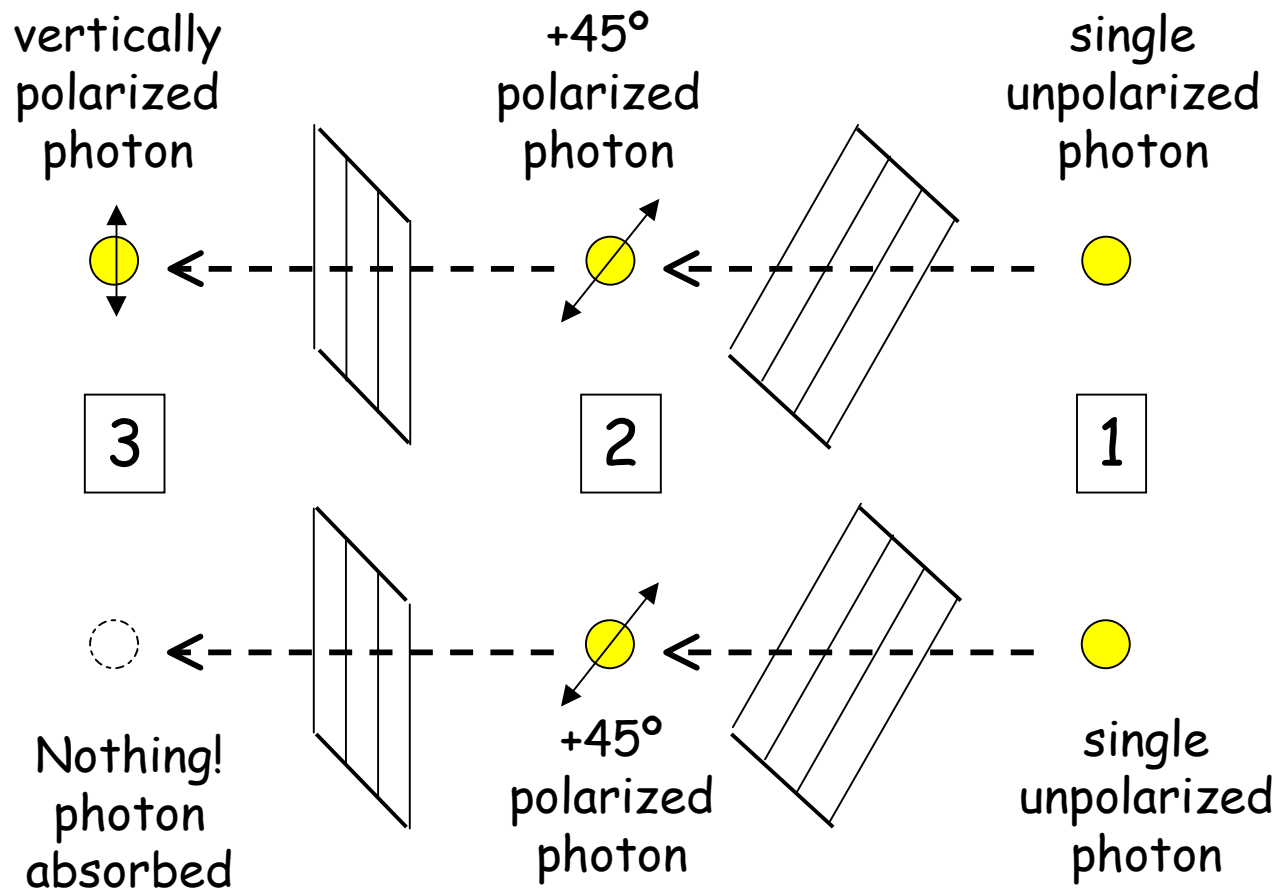
Are there any other approaches  
that might lead us to **provable**  
**computational security**?



# Quantum Physics?



The two equally likely outcomes of an experiment:



## Quantum Cryptography

is more accurately called

## Quantum Key Distribution

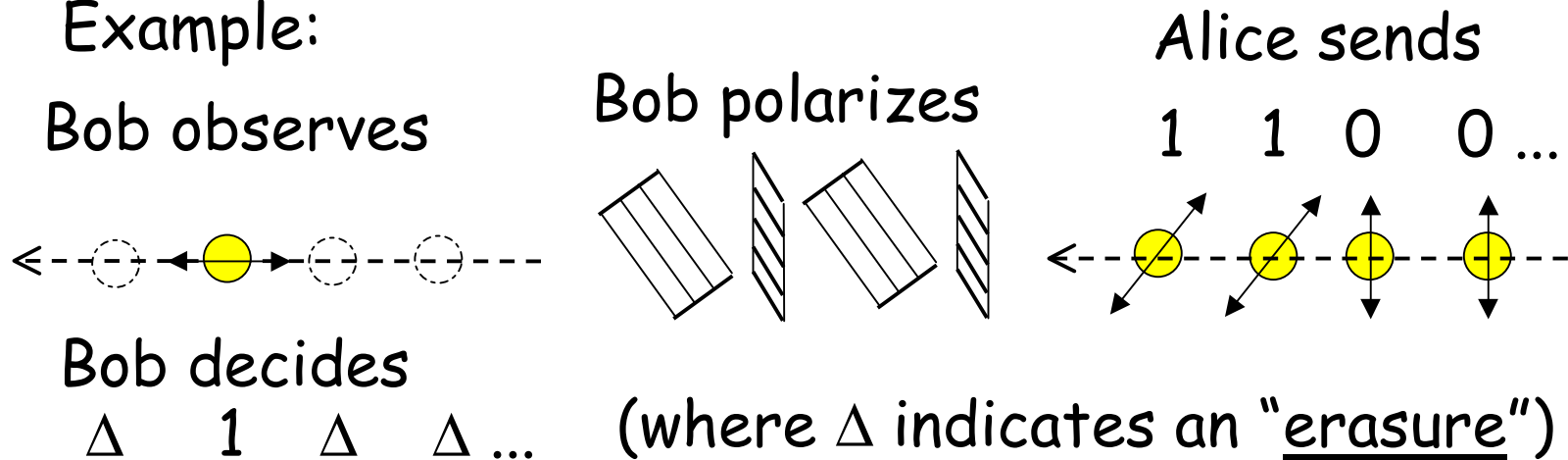
and even more accurately called

## Quantum Key Agreement

Quantum Key Agreement refers to schemes in which **two parties reach agreement on a random key (chosen by nature)** in such a way that an eavesdropper will obtain no information about this key and, moreover, the presence of the **eavesdropper** will be detected if the eavesdropping is done for an extended period.

The basic idea: Alice transmits a random sequence by sending a  $+45^\circ$  polarized photon to represent a 1 and a vertically polarized photon to represent a 0. Bob randomly chooses between a horizontal polarizer and a  $-45^\circ$  polarizer to detect each photon he receives.

Example:



- Bob's decisions (non-erasures) will never be wrong
- Eavesdropper will cause errors with probability  $1/4$ .

Alice and Bob need a protocol, which includes the use of erasure-correcting codes, to reach agreement on a key of a specified size in such a way that the eavesdropper is kept in the dark with high probability—unless the eavesdropper “listens” to a substantial fraction of the photons in which case the eavesdropper will be detected through the error probability that Alice and Bob observe during the performance of their protocol.

The eavesdropper can successfully **deny service** to Alice and Bob by listening to all transmissions.

Is it easier to believe in the realizability of quantum-rules for single-photon transmission than it is to believe in the realizability of a BSS?

Essentially there has been **zero progress** toward a **mathematical theory of cryptography for computational security**, whether in secret-key or in public-key cryptography!

**Today** almost no one works on the problem of developing provably computationally-secure systems!

**Today** almost everyone plies the **art** of cryptography, generating more and more schemes that nobody can prove are computationally secure.

Developing **provable computational security** is a wide-open area of research, success in which could have enormous practical consequences.

It might not be easy!

**“Problems worthy of attack,  
prove their worth by hitting back!”**  
Piet Hein

