What Can Be Proved About Security?

Palash Sarkar

Applied Statistics Unit
Indian Statistical Institute, Kolkata
India
palash@isical.ac.in

Centre for Artificial Intelligence and Robotics
Bengaluru
23rd February 2012

The Context

The scariest thought for a designer of a cryptosystem is that it will be broken.

The Context

The scariest thought for a designer of a cryptosystem is that it will be broken.

- It is enough to give a designer sleepless nights.
- For an actually deployed system a designer remains perpetually uneasy about security.

The Context

The scariest thought for a designer of a cryptosystem is that it will be broken.

- It is enough to give a designer sleepless nights.
- For an actually deployed system a designer remains perpetually uneasy about security.

A designer needs assurance that the system is indeed secure.

How to obtain an assurance that a cryptosystem is secure?

Obtaining Security Assurance

Conventional approach:

- Get a number of people to investigate the security.
- If all persons fail in their cryptanalytic attempts, then that gives some confidence that the system is secure.

Obtaining Security Assurance

Conventional approach:

- Get a number of people to investigate the security.
- If all persons fail in their cryptanalytic attempts, then that gives some confidence that the system is secure.

However:

- What if the people looking at the system
  - are not ‘good’ enough?
  - are not motivated enough?
  - lack adequate training and knowledge of cryptanalysis?
- What if somebody with better ability breaks the system sometime in the future?

Obtaining Security Assurance

Desirable: Obtain a ‘proof’ that a system is secure.

- Proofs/arguments that a system withstands all *known* attacks.
  - This is necessary and provides valuable information about the security of a system.

Obtaining Security Assurance

Desirable: Obtain a ‘proof’ that a system is secure.

- Proofs/arguments that a system withstands all *known* attacks.
  - This is necessary and provides valuable information about the security of a system.

  But, the system may be vulnerable to hitherto undiscovered attacks.

Obtaining Security Assurance

Desirable: Obtain a ‘proof’ that a system is secure.

- Proofs/arguments that a system withstands all *known* attacks.
  - This is necessary and provides valuable information about the security of a system.

  But, the system may be vulnerable to hitherto undiscovered attacks.
- A proof that a system is secure against ‘*all*’ possible attacks.

Obtaining Security Assurance

Desirable: Obtain a ‘proof’ that a system is secure.

- Proofs/arguments that a system withstands all *known* attacks.
  - This is necessary and provides valuable information about the security of a system.

  But, the system may be vulnerable to hitherto undiscovered attacks.
- A proof that a system is secure against ‘*all*’ possible attacks.
  - That would be great!
  - It would guard against current *and future* human efforts.

Obtaining Security Assurance

Desirable: Obtain a ‘proof’ that a system is secure.

- Proofs/arguments that a system withstands all *known* attacks.
  - This is necessary and provides valuable information about the security of a system.

  But, the system may be vulnerable to hitherto undiscovered attacks.
- A proof that a system is secure against ‘*all*’ possible attacks.
  - That would be great!
  - It would guard against current *and future* human efforts.

Research on provable security is an attempt to reach this utopia.

Obtaining Security Assurance

Desirable: Obtain a 'proof' that a system is secure.

- Proofs/arguments that a system withstands all *known* attacks.
  - This is necessary and provides valuable information about the security of a system.

  But, the system may be vulnerable to hitherto undiscovered attacks.
- A proof that a system is secure against '*all*' possible attacks.
  - That would be great!
  - It would guard against current *and future* human efforts.

Research on provable security is an attempt to reach this utopia.

This approach to security assurance should be carried out in conjuction with the conventional approach.

Shannon's Notion of Perfect Secrecy

Entropy. Let X be a random variable with distribution $(p_0, \ldots, p_\ell)$.

$$H(X) \stackrel{\Delta}{=} -\sum p_i \log_2 p_i.$$

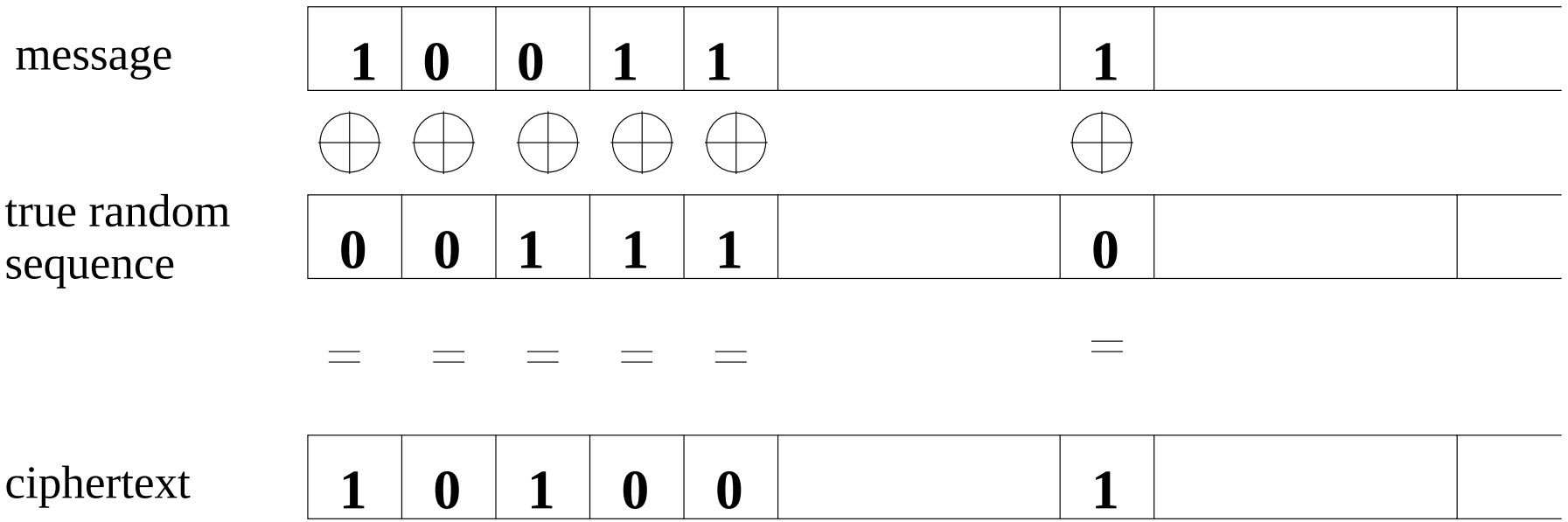
Conditional entropy.

$$H(Y|X) = \Pr[X = 0]H(Y|X = 0) + \Pr[X = 1]H(Y|X = 1).$$

Perfect Secrecy.

$$H(M) = H(M|C).$$

Vernam's One-Time Pad



Perfect Secrecy of One-Time Pad

For $a, b \in \{0, 1\}$,

$$
\begin{aligned}
\Pr[M_i = a | C_i = b] &= \frac{\Pr[M_i = a \wedge C_i = b]}{\Pr[C_i = b]} \\
&= \frac{\Pr[M_i = a, M_i \oplus K_i = b]}{\Pr[M_i \oplus K_i = b]} \\
&= \frac{\Pr[M_i = a, K_i = a \oplus b]}{\Pr[M_i = 0, K_i = b] + \Pr[M_i = 1, K_i = 1 \oplus b]} \\
&= \frac{\Pr[M_i = a] \times \Pr[K_i = a \oplus b]}{\Pr[M_i = 0]\Pr[K_i = b] + \Pr[M_i = 1]\Pr[K_i = 1 \oplus b]} \\
&= \frac{\frac{1}{2}\Pr[M_i = a]}{\frac{1}{2} \times (\Pr[M_i = 0] + \Pr[M_i = 1])} \\
&= \frac{\frac{1}{2}\Pr[M_i = a]}{\frac{1}{2}} \\
&= \Pr[M_i = a].
\end{aligned}
$$

Key Features of this Approach

- A heavy dose of randomness and probability.

Key Features of this Approach

- A heavy dose of randomness and probability.
- Models adversarial resources.
  - Computationally unbounded adversary having access to ciphertext.

Key Features of this Approach

- A heavy dose of randomness and probability.
- Models adversarial resources.
  - Computationally unbounded adversary having access to ciphertext.
- Models adversarial goal.
  - Gain some information about the message.

Key Features of this Approach

- A heavy dose of randomness and probability.
- Models adversarial resources.
  - Computationally unbounded adversary having access to ciphertext.
- Models adversarial goal.
  - Gain some information about the message.
- Provides a precise definition of what is meant by security.
  - Entropy of the message does not decrease when conditioned upon the ciphertext.

Key Features of this Approach

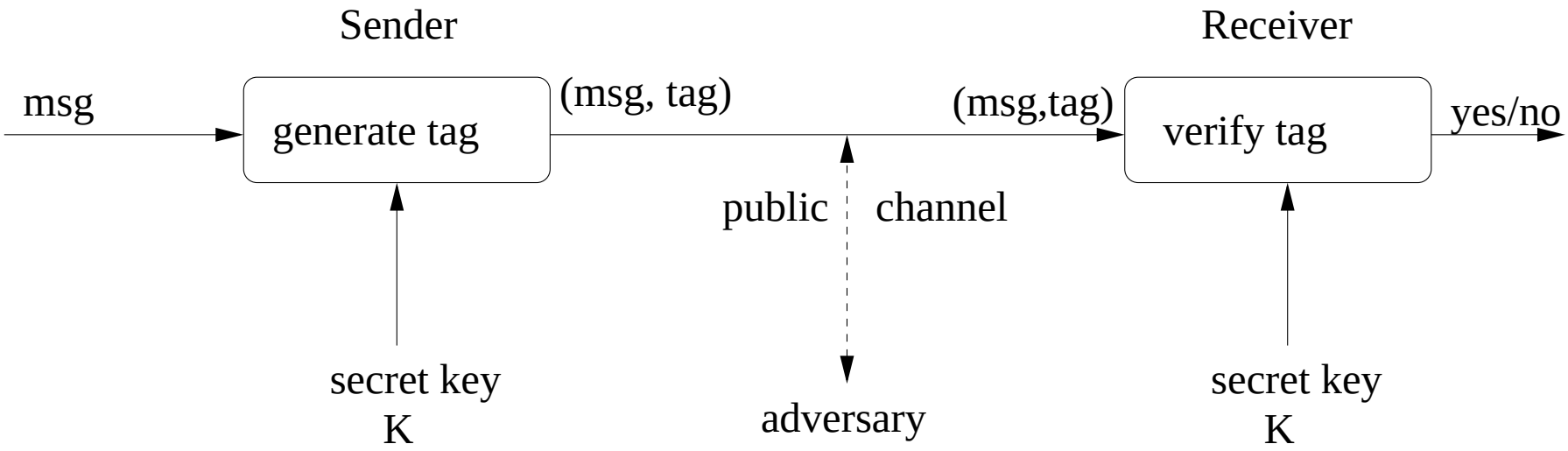
- A heavy dose of randomness and probability.
- Models adversarial resources.
  - Computationally unbounded adversary having access to ciphertext.
- Models adversarial goal.
  - Gain some information about the message.
- Provides a precise definition of what is meant by security.
  - Entropy of the message does not decrease when conditioned upon the ciphertext.
- Provides a proof that the system in question satisfies the definition.

Key Features of this Approach

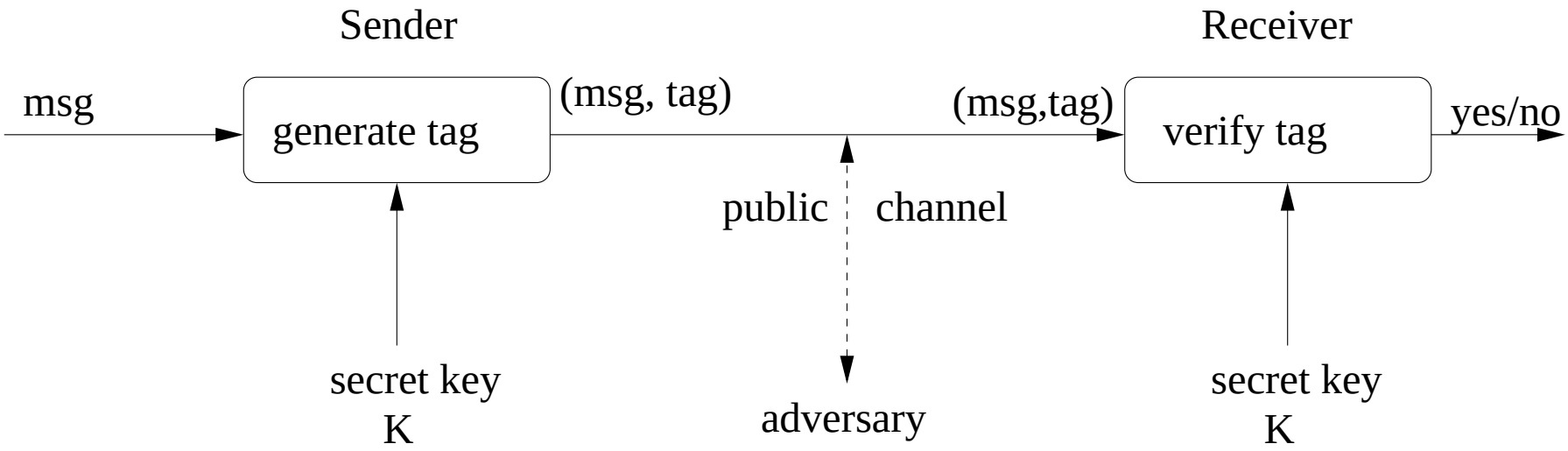
- A heavy dose of randomness and probability.
- Models adversarial resources.
  - Computationally unbounded adversary having access to ciphertext.
- Models adversarial goal.
  - Gain some information about the message.
- Provides a precise definition of what is meant by security.
  - Entropy of the message does not decrease when conditioned upon the ciphertext.
- Provides a proof that the system in question satisfies the definition.

These are the key features of all subsequent work on provable security.

Example: Symmetric Key Authentication

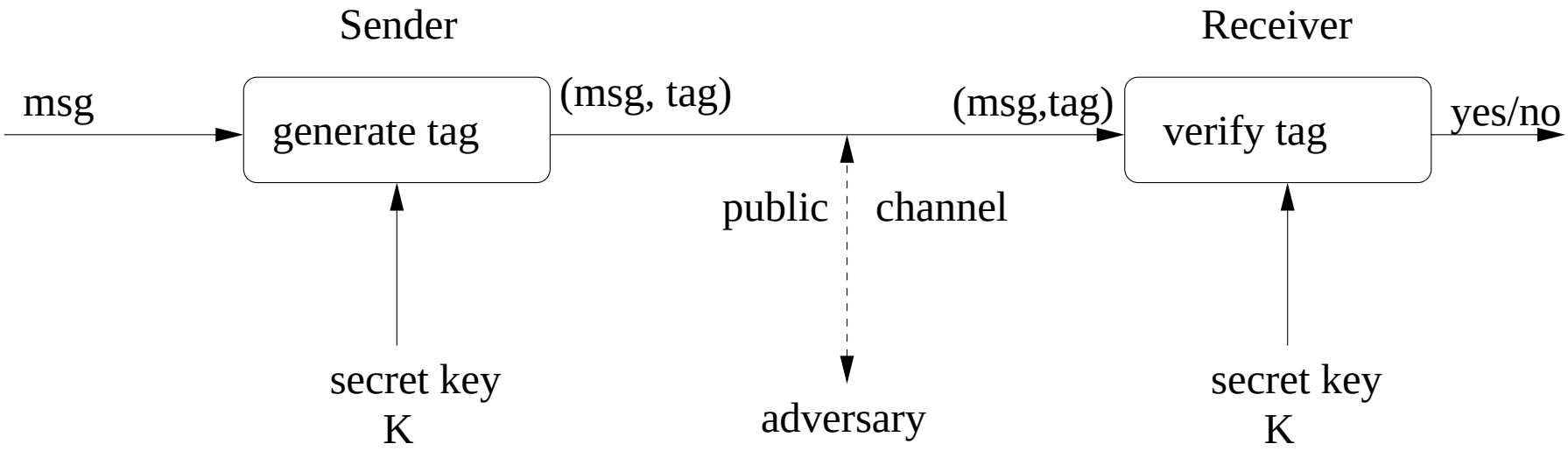


Example: Symmetric Key Authentication



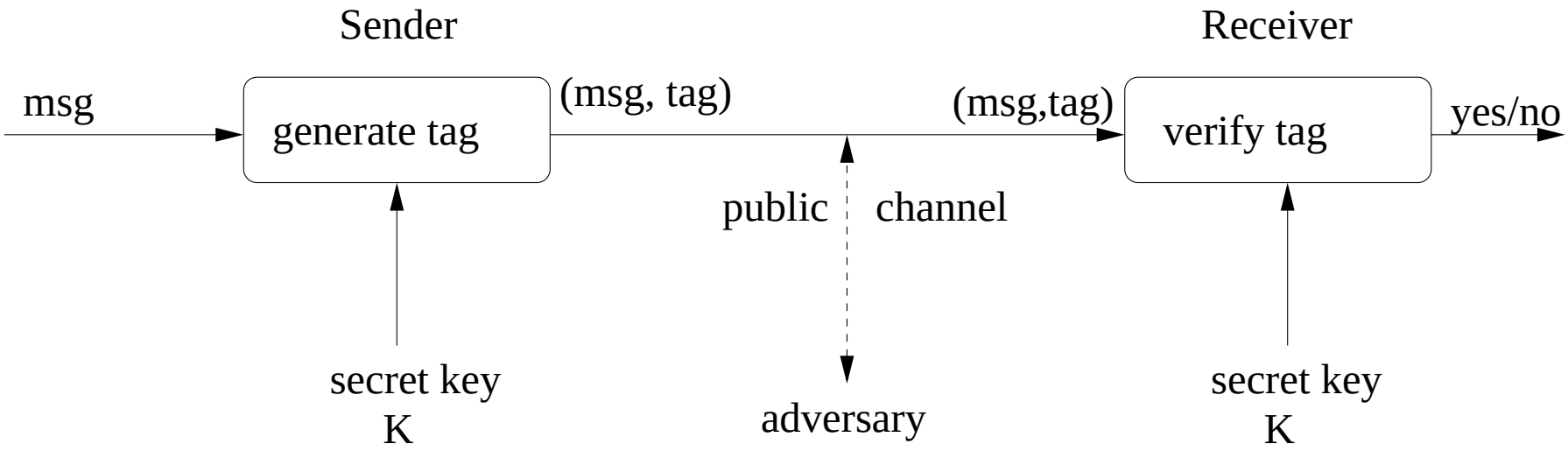
- **Adversarial Capability:**
  - Can listen to and modify information on the public channel.
  - Can obtain tags corresponding to chosen messages.

Example: Symmetric Key Authentication



- **Adversarial Capability:**
  - Can listen to and modify information on the public channel.
  - Can obtain tags corresponding to chosen messages.
- **Adversarial goal:**
  - To make the receiver accept a msg-tag pair not generated by the sender.

Example: Symmetric Key Authentication



- **Adversarial Capability:**
  - Can listen to and modify information on the public channel.
  - Can obtain tags corresponding to chosen messages.
- **Adversarial goal:**
  - To make the receiver accept a msg-tag pair not generated by the sender.
- **Adversarial Success:** a random event.
  - Measured by the probability that the receiver accepts a forgery.

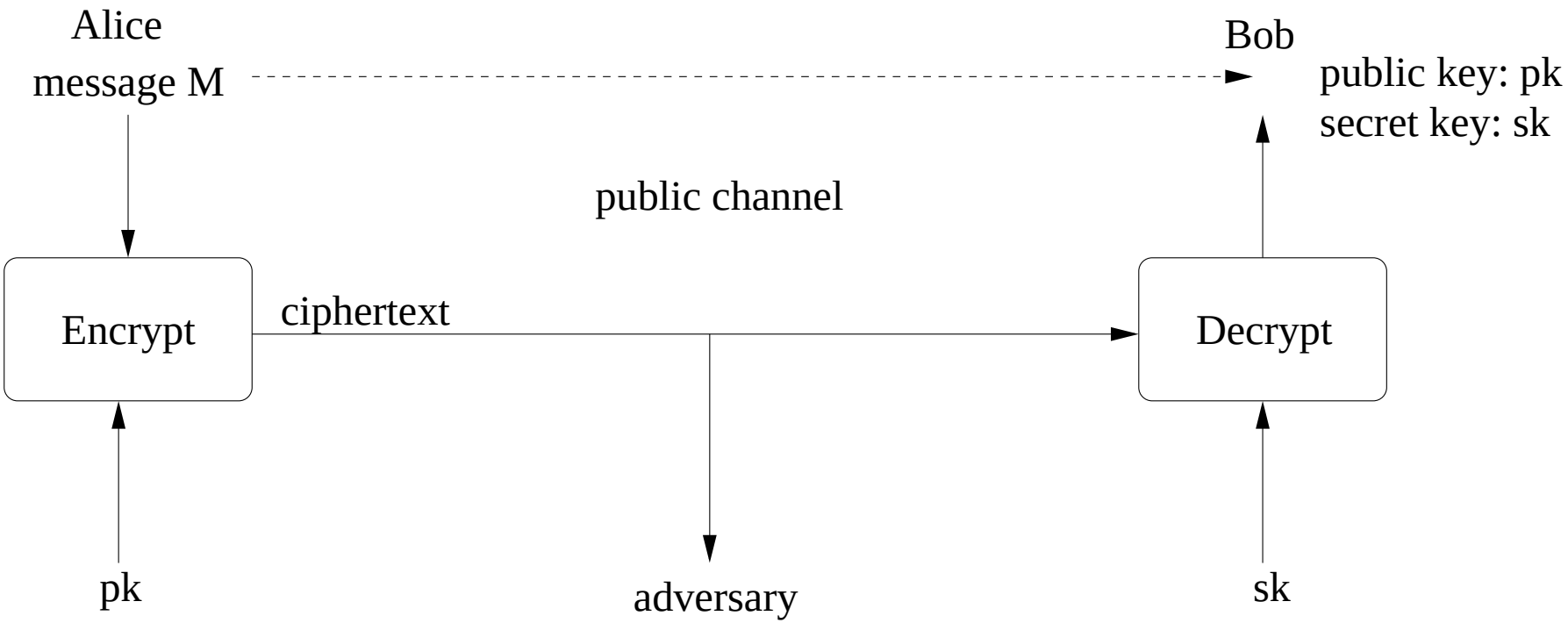
Provably Secure Authentication

- There are several known authentication schemes in the literature.
- Such schemes are usually built from more basic primitives such as block ciphers, stream ciphers and hash functions.

Provably Secure Authentication

- There are several known authentication schemes in the literature.
- Such schemes are usually built from more basic primitives such as block ciphers, stream ciphers and hash functions.
- For most schemes, there is usually a proof showing that an adversary's success probability is low.
- The analysis is based upon an appropriate assumption on the underlying primitive.
  - E.g.: a block cipher is computationally indistinguishable from a uniform random permutation.
- The proof is a reduction: If the underlying primitive is secure, then so is the authentication scheme.
- The bulk of such proofs is usually a rather involved probability analysis.

Example: Public Key Encryption



Security Definition

The basic idea behind the definition is to capture the notion of

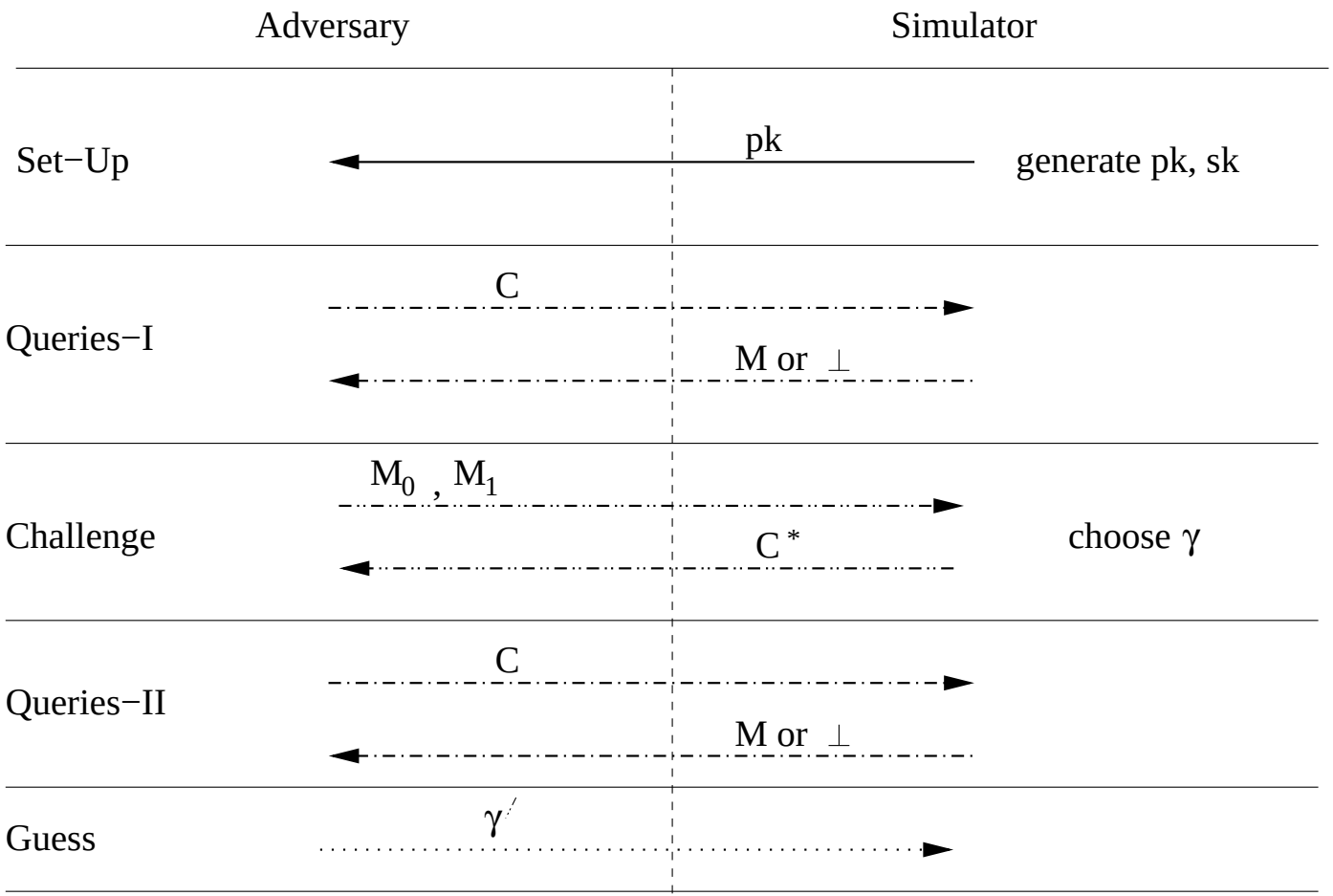
computational indistinguishability.

- Suppose M_0 and M_1 are two messages.
- Let $\mathcal{C}_0$ and $\mathcal{C}_1$ be the set of possible ciphertexts that can arise from M_0 and M_1.
- Suppose a bit b is chosen uniformly at random and C^* is chosen uniformly at random from $\mathcal{C}_b$.
- Given C^*, the task of the adversary is to determine the value of b.

The adversary is assumed to have black-box access to the decryption algorithm.

- It can get messages corresponding to chosen ciphertexts (other than C^*).

Security Model for PKE



Adversary's Advantage

For an adversary $\mathcal{A}$,

$$\mathsf{Adv}(\mathcal{A}) = \left| \Pr[\gamma = \gamma'] - \frac{1}{2} \right| .$$

- Resource constraints on $\mathcal{A}$:
  - bound on runtime,
  - bound on the number of oracle queries.
- $\mathsf{Adv}(t, q)$: maximum (supremum) of $\mathsf{Adv}(\mathcal{A})$, over all adversaries $\mathcal{A}$ running in time t and making q oracle queries.

Security Assurance

If

lower level primitives are secure
and some problem Π is computationally hard
(and some functions are assumed to be uniform random functions)

then

the main protocol is secure.

Structure of Proofs

$$\underline{\text{A Game Sequence}}$$
$$G_0,$$
$$G_1,$$
$$\vdots$$
$$G_k$$

- Let X_i be the event that $\gamma = \gamma'$ in Game G_i. We consider

$$\begin{aligned}
&\Pr[X_0],\\
&\Pr[X_0] - \Pr[X_1],\\
&\vdots\\
&\Pr[X_{k-1}] - \Pr[X_k]\\
&\Pr[X_k].
\end{aligned}$$

Structure of Proofs (contd.)

- G_0 is the game which defines the security of the protocol and so

$$\mathsf{Adv}(\mathcal{A}) = |\Pr[\gamma = \gamma'] - 1/2| = |\Pr[X_0] - 1/2|.$$

- G_k is designed such that the bit γ is statistically hidden from the adversary. So,

$$\Pr[X_k] = 1/2.$$

- Games G_{i-1} and G_i differ:
  - the difference is not too much;
  - the adversary should not be able to notice whether he is playing Game G_{i-1} or Game G_i.

Structure of Proofs (contd.)

- More precisely, $\Pr[X_{i-1}] - \Pr[X_i]$ is bounded above by
  - either a 'small' quantity;
  - or, the advantage of an adversary in breaking one of the smaller protocols;
  - or, the advantage of solving problem Π.

$$
\begin{aligned}
\mathrm{Adv}(\mathcal{A}) &= |\Pr[X_0] - 1/2| \\
&= |\Pr[X_0] - \Pr[X_k]| \\
&\leq |\Pr[X_0] - \Pr[X_1]| \\
&\quad + |\Pr[X_1] - \Pr[X_2]| \\
&\quad + \cdots \\
&\quad + |\Pr[X_{k-1}] - \Pr[X_k]|.
\end{aligned}
$$

Provably Secure Schemes

A cryptographic scheme has a security proof in an appropriate security model.

Provably Secure Schemes

A cryptographic scheme has a security proof in an appropriate security model.

Period

Provably Secure Schemes

A cryptographic scheme has a security proof in an appropriate security model.

Period

No more sleepless nights for the designer.

But the adversary never sleeps!

Provable Security: Limitations

Can we rely completely on the provable security approach?

- The security model may cover many attacks.
  But, is it possible to have a model which captures *all* possible *real world* attacks including those that can occur in the future?

Provable Security: Limitations

Can we rely completely on the provable security approach?

- The security model may cover many attacks.
  But, is it possible to have a model which captures *all* possible *real world* attacks including those that can occur in the future?
- Example: Side-channel attacks are not covered by the usual security definition of PKE.
- In recent years attempts have been made to model such attacks and obtain schemes which are also provably secure against them.

Provable Security: What do the Proofs Guarantee?

- All such proofs are reductions. They show that a scheme is secure if some problem is hard and/or other schemes are secure.
- The quantitative relation between the hardness of a problem and the security of a scheme is important.
  - If the relation is not tight, then how should the proof be viewed?
- The proof may go through only in one direction. Ability to solve the problem may not lead to an attack on the scheme.

Provable Security Versus Efficiency

- Suppose there is no proof that a scheme is secure and neither there is an attack on the scheme.
- Do we reject the scheme simply because it does not have an associated proof?
- It may be more efficient than schemes which have security proofs
  - In the real-world efficiency matters!

Future of Provable Security

Provable security is here to stay.

Conventional security analysis will also continue to be important.

Future of Provable Security

Provable security is here to stay.

Conventional security analysis will also continue to be important.

Having a security proof is not the only criterion for deploying a scheme. It is certainly an important and desirable criterion.

The proof, underlying assumptions and the security model of a provably secure scheme needs to be carefully analysed before deciding on deployment of the scheme.

Thank you for your attention!