Formal and Strong Security Definitions I

There are three kinds of lies:
small lies, big lies and statistics.

Sven Laur
swen@math.ut.ee

Helsinki University of Technology

Basic theoretical notions

Formal syntax of a cryptosystem I

Various domains associated with the cryptosystem:

$\mathcal{M}$ – a set of plausible messages (plaintexts);

$\mathcal{C}$ – a set of possible cryptograms (ciphertexts);

$\mathcal{R}$ – random coins used by the encryption algorithm.

Parameters used by the encryption and decryption algorithms:

pk – a public key (public knowledge needed to generate valid encryptions);

sk – a secret key (knowledge that allows to efficiently decrypt ciphertexts).

Formal syntax of a cryptosystem II

Algorithms that define a cryptosystem:

$\mathcal{G}$ – a randomised key generation algorithm;

$\mathcal{E}_{\mathsf{pk}}$ – a randomised encryption algorithm;

$\mathcal{D}_{\mathsf{sk}}$ – a deterministic decryption algorithm.

The key generation algorithm $\mathcal{G}$ outputs a random key pair $(\mathsf{pk}, \mathsf{sk})$.

The encryption algorithm is an efficient mapping $\mathcal{E}_{\mathsf{pk}} : \mathcal{M} \times \mathcal{R} \to \mathcal{C}$.

The decryption algorithm is an efficient mapping $\mathcal{D}_{\mathsf{sk}} : \mathcal{C} \to \mathcal{M}$.

A cryptosystem must be functional

$$\forall(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{G},\ \forall m \in \mathcal{M},\ \forall r \in \mathcal{R} : \quad \mathcal{D}_{\mathsf{sk}}(\mathcal{E}_{\mathsf{pk}}(m; r)) = m.$$

When is a cryptosystem secure?

It is rather hard to tell when a cryptosystem is secure. Instead people often specify when a cryptosystem is broken:

- *Complete key recovery.* Given pk and $\mathcal{E}_{\mathsf{pk}}(m_1), \ldots, \mathcal{E}_{\mathsf{pk}}(m_n)$, the adversary deduces sk in a *feasible* time with a *reasonable* probability.
- *Complete plaintext recovery.* Given pk and $\mathcal{E}_{\mathsf{pk}}(m_1), \ldots, \mathcal{E}_{\mathsf{pk}}(m_n)$, the adversary is able to recover m_i in a *feasible* time with a *reasonable* probability.
- *Partial plaintext recovery.* Given pk and $\mathcal{E}_{\mathsf{pk}}(m_1), \ldots, \mathcal{E}_{\mathsf{pk}}(m_n)$, the adversary is able to recover a part of m_i in a *feasible* time with a *reasonable* probability.

The list is not complete and neither can never be completed!

Semantic security

Shaft Goldwasser and Silvio Micali, *Probabilistic Encryption & How To Play Mental Poker Keeping Secret All Partial Information*, 1982.

> A Public Key Cryptosystem is ε secure if an adversary does not have an ε advantage in evaluating, given the ciphertext, any easy to compute predicate relative to the cleartext.

Contemporary treatment of semantic security:

- Mihir Bellare, Anand Desai, E. Jokipii and Phillip Rogaway, *A Concrete Security Treatment of Symmetric Encryption*, 1997.
- Mihir Bellare, Anand Desai, David Pointcheval and Phillip Rogaway, *Relations among Notions of Security for Public-Key Encryption Schemes*, 1998.

IND-CPA security

Malice is good in breaking security of a cryptosystem $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ if Malice can distinguish two experiments (hypothesis testing):

Experiment Exp_0	Experiment Exp_1
1. $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{G}$	1. $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{G}$
2. $(m_0, m_1, \sigma) \leftarrow \mathsf{Malice}(\mathsf{pk})$	2. $(m_0, m_1, \sigma) \leftarrow \mathsf{Malice}(\mathsf{pk})$
3. $\mathsf{guess} \leftarrow \mathsf{Malice}(\sigma, \mathcal{E}_{\mathsf{pk}}(m_0))$	3. $\mathsf{guess} \leftarrow \mathsf{Malice}(\sigma, \mathcal{E}_{\mathsf{pk}}(m_1))$

with a *non-negligible** advantage

$$\mathsf{Adv}(\mathsf{Malice}) = \frac{1}{2} \cdot \left| \underbrace{\Pr\left[\mathsf{guess} = 0 | \mathsf{Exp}_0\right]}_{\text{True positives}} - \underbrace{\Pr\left[\mathsf{guess} = 0 | \mathsf{Exp}_1\right]}_{\text{False positives}} \right|$$

Bit-guessing game with a fair coin

Consider Protocol 14.1 in Mao's book:

1. $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{G}$
2. $(m_0, m_1, \sigma) \leftarrow \mathsf{Malice}(\mathsf{pk})$ where σ denotes advice, e.g. pk.
3. Oracle $\mathcal{O}$ flips a fair coin $b \leftarrow \{0, 1\}$ and sets $c \leftarrow \mathcal{E}_{\mathsf{pk}}(m_b)$.
4. $\mathsf{guess} \leftarrow \mathsf{Malice}(\sigma, c)$

$$
\begin{aligned}
\Pr\left[\mathsf{guess} = b\right] &= \Pr\left[b = 0\right] \Pr\left[\mathsf{guess} = 0 | b = 0\right] + \Pr\left[b = 1\right] \Pr\left[\mathsf{guess} = 1 | b = 1\right] \\
&= \frac{1}{2} \cdot \Pr\left[\mathsf{guess} = 0 | \mathsf{Exp}_0\right] + \frac{1}{2} \cdot \left(1 - \Pr\left[\mathsf{guess} = 0 | \mathsf{Exp}_1\right]\right) \\
&= \frac{1}{2} \pm \mathsf{Adv}(\mathsf{Malice})
\end{aligned}
$$

Bit-guessing game with a biased coin*

Consider the bit-guessing game when the coin is biased $\Pr[b = 1] = \frac{3}{4}$.

Show that the probability of correct answer is in the range

$$\frac{1}{4} - \frac{1}{2} \cdot \mathsf{Adv}(\mathsf{Malice}) \leq \Pr[\mathsf{guess} = b] \leq \frac{3}{4} + \frac{1}{2} \cdot \mathsf{Adv}(\mathsf{Malice})$$

Give an interpretation to the formula.

Is there any way to "cleverly" use subroutine Malice so that

$$\Pr[\mathsf{guess} = b] = \frac{3}{4} + \frac{1}{2} \cdot \mathsf{Adv}(\mathsf{Malice})?$$

IND-CPA $\implies$ Semantic security

Let $\pi : \mathcal{M} \rightarrow \{0,1\}$ be a predicate such that $\Pr\left[m \leftarrow \mathcal{M} : \pi(m) = 1\right] = \frac{1}{2}$.

If Charlie can efficiently and correctly guess $\pi(m)$ given only pk and $\mathcal{E}_{\mathsf{pk}}(m)$:

$$\mathsf{Adv}^{\mathsf{guess}}(\mathsf{Charlie}) = \Pr\left[\begin{array}{l}(\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{G}, m \leftarrow \mathcal{M} : \\ \mathsf{Charlie}(\mathsf{pk}, \mathcal{E}_{\mathsf{pk}}(m)) = \pi(m)\end{array}\right] - \frac{1}{2} \geq 0$$

then we can construct Malice:

1. Malice chooses $m_0, m_1 \leftarrow \mathcal{M}$ randomly.
2. Given $c = \mathcal{E}_{\mathsf{pk}}(m_b)$, Malice runs Charlie:
   - If $\mathsf{Charlie}(\mathsf{pk}, c) = \pi(m_0)$ return 0
   - Else return 1.

How well does Malice perform?

Evidently, we can write

$$\Pr\left[\mathsf{guess}=0|\mathsf{Exp}_0\right]=\Pr\left[\begin{array}{l}(\mathsf{pk},\mathsf{sk})\leftarrow\mathcal{G},m_0,m_1\leftarrow\mathcal{M}:\\ \mathsf{Charlie}(\mathsf{pk},\mathcal{E}_{\mathsf{pk}}(m_0))=\pi(m_0)\end{array}\right]$$

$$\Pr\left[\mathsf{guess}=0|\mathsf{Exp}_1\right]=\Pr\left[\begin{array}{l}(\mathsf{pk},\mathsf{sk})\leftarrow\mathcal{G},m_0,m_1\leftarrow\mathcal{M}:\\ \mathsf{Charlie}(\mathsf{pk},\mathcal{E}_{\mathsf{pk}}(m_1))=\pi(m_0)\end{array}\right]$$

and thus

$$\begin{aligned}2\mathsf{Adv}(\mathsf{Malice})&=\left|\frac{1}{2}+\mathsf{Adv}^{\mathsf{guess}}(\mathsf{Charlie})-\Pr\left[\mathsf{Charlie}(\mathsf{pk},\mathcal{E}_{\mathsf{pk}}(m_1))=\pi(m_0)\right]\right|\\&=\mathsf{Adv}^{\mathsf{guess}}(\mathsf{Charlie})\end{aligned}$$

since for fixed m_1, we have always $\Pr\left[\mathsf{Charlie}(\mathsf{pk},\mathcal{E}_{\mathsf{pk}}(m_1))=\pi(m_0)\right]=\frac{1}{2}$.

IND-CPA $\Longrightarrow$ Semantic security

Why does IND-CPA security imply semantic security w.r.t. π?

Why π must be efficiently computable?

Extend the proof for the general case where π is not a balanced predicate*.

What if Charlie can predict a function $f : \mathcal{M} \to \mathbb{N}$ from pk and $\mathcal{E}_{\mathsf{pk}}(m)$?

Extend the proof for the general case where Charlie predicts f^*.

How much time can Malice spend?

Usually, it is assumed that Malice uses a probabilistic polynomial time algorithm to launch the attack. What does it mean?

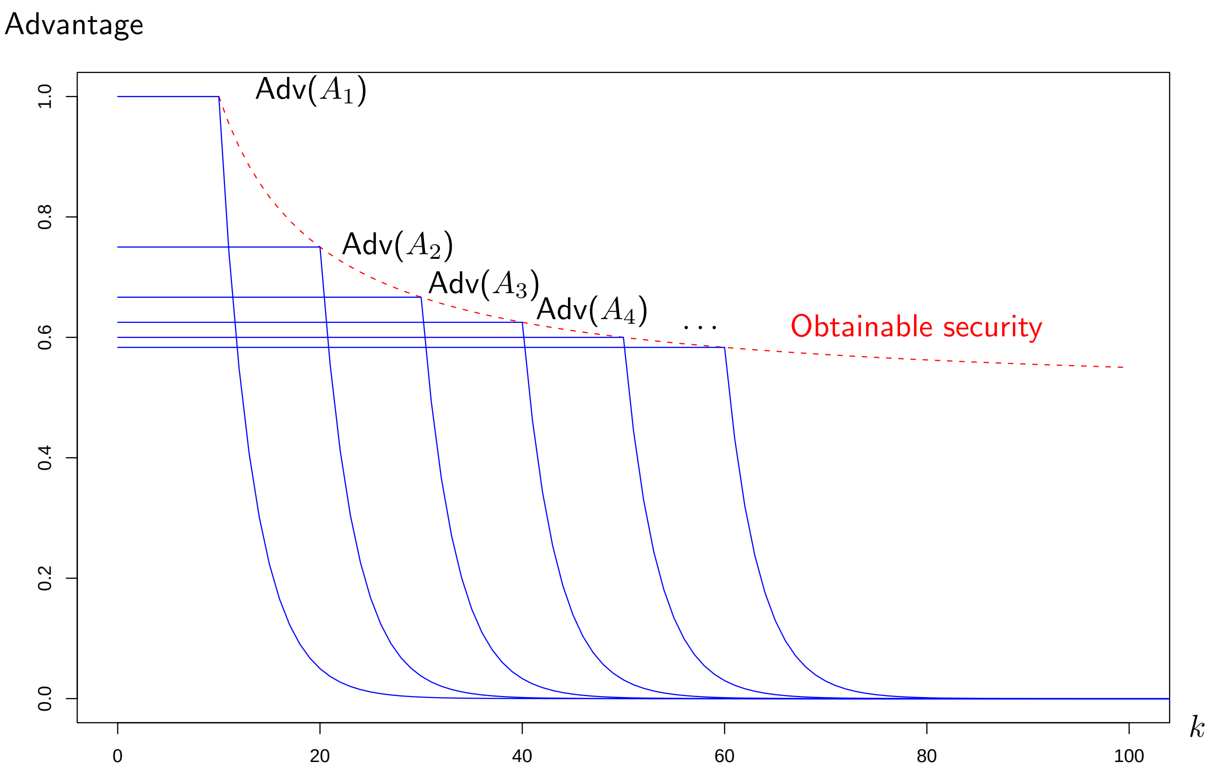
Example
1994 – 426 bit RSA challenge broken.
2003 – 576 bit RSA challenge broken.
2005 – 640 bit RSA challenge broken.

Instead of a concrete encryption scheme RSA is a family of cryptosystems and Malice can run algorithm polynomial in the length k of RSA modulus.

Negligible advantage means that the advantage decreases faster than k^{-c} for any $c > 0$.

A concrete example

For simplicity, imagine that Malice runs algorithms that finish in time k^5.



Uniform *vs* non-uniform security

For each polynomial-time algorithm A_i the advantage was negligible:
$\Longrightarrow$ scheme is secure against polynomial *uniform* adversaries.

If Malice chooses a good algorithm for each k separately
$\Longrightarrow$ she breaks the scheme with advantage $\frac{1}{2}$;
$\Longrightarrow$ scheme is insecure against polynomial *non-uniform* adversaries.

In practice, each adversary has limited resources
$\Longrightarrow$ Given time t, Malice should not achieve $\mathsf{Adv}(\mathsf{Malice}) \geq \varepsilon_{\mathsf{critical}}$.

If scheme is secure against non-uniform adversaries then for large k:
$\Longrightarrow$ $\mathsf{Adv}(\mathsf{Malice}) \leq \varepsilon_{\mathsf{critical}}$ for all t time algorithms;
$\Longrightarrow$ the scheme is still efficiently implementable.

Is non-uniform security model adequate in practice*?

Consider the case of browser certificates:

- Several Verisign certificates have been issued in 1996–1998.
- As a potential adversary knows pk, he can design a special crack algorithm for that pk only. He does not care about other values of pk.
- Maybe a special bit pattern of $N = pq$ allows more efficient factorisation?

Why can't we fix pk in the non-uniform model?

Is there a model that describes reality without problems*?

Does security against (non-)uniform adversaries *heuristically* imply security in real applications*?

Concrete examples

Commutative cryptosystems

A cryptosystem $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ is commutative if for any valid public keys $\mathsf{pk}_\mathsf{A}, \mathsf{pk}_\mathsf{B}$

$$\forall m \in \mathcal{M} : \quad \mathcal{E}_{\mathsf{pk}_\mathsf{A}}(\mathcal{E}_{\mathsf{pk}_\mathsf{B}}(m)) = \mathcal{E}_{\mathsf{pk}_\mathsf{B}}(\mathcal{E}_{\mathsf{pk}_\mathsf{A}}(m)).$$

In particular it implies

$$m = \mathcal{D}_{\mathsf{sk}_\mathsf{A}}(\mathcal{D}_{\mathsf{sk}_\mathsf{B}}(\mathcal{E}_{\mathsf{pk}_\mathsf{A}}(\mathcal{E}_{\mathsf{pk}_\mathsf{B}}(m)))) = \mathcal{D}_{\mathsf{sk}_\mathsf{B}}(\mathcal{D}_{\mathsf{sk}_\mathsf{A}}(\mathcal{E}_{\mathsf{pk}_\mathsf{B}}(\mathcal{E}_{\mathsf{pk}_\mathsf{A}}(m)))).$$

The latter allows to swap the order of encryption and decryption operations.

Mental poker protocol

1. Alice sends randomly shuffled encryptions $\mathcal{E}_{\mathsf{pk_A}}(\spadesuit 2), \ldots, \mathcal{E}_{\mathsf{pk_A}}(\heartsuit \mathrm{A})$.
2. Bob chooses randomly c_A, c_B and sends $c_A, \mathcal{E}_{\mathsf{pk_B}}(c_B)$ to Alice.
3. Alice sends $\mathcal{D}_{\mathsf{sk_A}}(\mathcal{E}_{\mathsf{pk_B}}(c_B))$ to Bob and locally outputs $\mathcal{D}_{\mathsf{sk_A}}(c_A)$.
4. Bob outputs locally $\mathcal{D}_{\mathsf{sk_B}}(\mathcal{D}_{\mathsf{sk_A}}(\mathcal{E}_{\mathsf{pk_B}}(c_B))) = \mathcal{D}_{\mathsf{sk_A}}(c_B)$.
5. Alice sends her $\mathsf{pk_A}$ to Bob. Bob sends his $\mathsf{pk_B}$ to Alice.

RSA with shared modulus $N = pq$, and keys $(\mathsf{pk_A}, \mathsf{sk_A}) = (e_A, d_A)$ and $(\mathsf{pk_B}, \mathsf{sk_B}) = (e_B, d_B)$ such that

$$e_A d_A = 1 \mod \phi(N) \qquad e_B d_B = 1 \mod \phi(N)$$

is insecure after Step 5. **Why?**

Attacks against mental poker game

Recall that RSA encryption preserves quadratic residuocity and both parties can compute it. Leaking residuocity can give an edge to Bob.

Brute force attack. Let $\spadesuit 2, \ldots, \heartsuit$A be encoded as $1, \ldots, 52$. Then corresponding encryptions are $1, 2^{e_A}, \ldots, 56^{e_A}$ modulo N. Obviously,

$$2^{e_A} \cdot 2^{e_A} = 4^{e_A} \mod N, \quad \ldots, \quad 7^{e_A} \cdot 7^{e_A} = 49^{e_A} \mod N$$

and Bob can with high probability separate encryptions of $2, \ldots, 7$.

Similar connections allow Bob to reveal most of the cards.

There are completely insecure encodings for the cards
$\Longrightarrow$ vanilla RSA is not applicable for secure encryption;
$\Longrightarrow$ vanilla RSA is not IND-CPA secure;

Goldwasser-Micali cryptosystem

Famous conjecture. Let N be a large RSA modulus. Then without factorisation of N it is infeasible to determine whether a random $c \in J_N(1)$ is a quadratic residue or not.

Key generation. Generate safe primes $p, q \in \mathbb{P}$ and choose quadratic non-residue $y \in J_N(1)$ modulo $N = pq$. Set pk $= (n, y)$, sk $= (p, q)$.

Encryption. First choose a random $x \leftarrow \mathbb{Z}_N^*$ and then compute

$$\mathcal{E}_{\mathsf{pk}}(0) = x^2 \mod N \quad \text{and} \quad \mathcal{E}_{\mathsf{pk}}(1) = yx^2 \mod N.$$

Decryption. Given c, compute $c_1 \mod p$ and $c_2 \mod q$ and use Euler's criterion to test whether c is a quadratic residue or not.

ElGamal cryptosystem

Combine the Diffie-Hellman key exchange protocol

Alice		**Bob**
$x \leftarrow \mathbb{Z}_{\lvert G \rvert}$	$\xrightarrow{y=g^x}$	$k \leftarrow \mathbb{Z}_{\lvert G \rvert}$
	$\xleftarrow{g^k}$	
$g^{xk} = (g^k)^x$		$g^{xk} = (g^x)^k$

with one-time pad using multiplication in $G = \langle g \rangle$ as encoding rule

$$\mathcal{E}_{\mathsf{pk}}(m) = (g^k, m \cdot g^{xk}) = (g^k, m \cdot y^k) \qquad \text{for all elements } m \in G$$

with a public key $\mathsf{pk} = y = g^x$ and a secret key $\mathsf{sk} = x$.

Decisional Diffie-Hellman Assumption (DDH)

DDH Assumption. For a fixed group G, Charlie can distinguish experiments

Exp_0	Exp_1
1. $x, k \leftarrow \mathbb{Z}_q,\ q = \lvert G \rvert$ 2. guess $\leftarrow$ Charlie(g, g^x, g^k, g^{xk})	1. $x, k, c \leftarrow \mathbb{Z}_q,\ q = \lvert G \rvert$ 2. guess $\leftarrow$ Charlie(g, g^x, g^k, g^c)

with a negligible advantage Adv(Charlie).

Obviously, the Diffie-Hellman key exchange protocol is secure under the DDH $\Longleftarrow$ we can change g^{xk} with g^c and Charlie cannot tell the difference.

If the Diffie-Hellman key exchange protocol is secure
$\Longrightarrow$ ElGamal is secure, as the one-time pad is unbreakable.

DDH $\Longrightarrow$ ElGamal is IND-CPA

Let Malice be good in IND-CPA game. Now Charlie given (g, g^x, g^k, z):

1. Set pk $= g^x$ and $(m_0, m_1, \sigma) \leftarrow$ Malice(pk).
2. Toss a fair coin $b \leftarrow \{0, 1\}$ and set $c = (g^k, m_b z)$.
3. Get guess $\leftarrow$ Malice(σ, c).
4. If guess $= b$ return 0 else output 1.

We argue that this is a good strategy to win DDH game.

Charlie's advantage in DDH game

Observe

$$\Pr\left[\mathsf{Charlie} = 0 | \mathsf{Exp}_0\right] = \Pr\left[\mathsf{Success\ in\ bit\ guessing\ game}\right] = \frac{1}{2} \pm \mathsf{Adv}(\mathsf{Malice})$$

$$\Pr\left[\mathsf{Charlie} = 0 | \mathsf{Exp}_1\right] = \Pr\left[\mathsf{Guess}\ b\ \mathsf{given\ a\ random\ cryptogram}\right] = \frac{1}{2}$$

and we get

$$\begin{aligned}\mathsf{Adv}(\mathsf{Charlie}) &= \frac{1}{2} \cdot \left|\Pr\left[\mathsf{Charlie} = 0 | \mathsf{Exp}_0\right] - \Pr\left[\mathsf{Charlie} = 0 | \mathsf{Exp}_1\right]\right| \\ &= \frac{1}{2} \cdot \mathsf{Adv}(\mathsf{Malice})\end{aligned}$$

Therefore good attack against IND-CPA game implies good attack against DDH game.

Why some instantiations of ElGamal fail?

If the message $m \notin G$ then mg^{xk} is not one-time pad, for example

$$G = \langle 2 \mod 6 \rangle \qquad \Longrightarrow \qquad m2^{xk} = m \mod 2$$

and a single bit of information is always revealed.

Fix a generator of $g \in \mathbb{Z}_p^*$ for large $p \in \mathbb{P}$ such that DDH holds.
If public key $y = g^x$ is quadratic residue (QR), then y^k is also QR.

$$\boxed{m \text{ is QR if and only if } my^k \text{ is QR}}$$

Fix I. Choose $g \in$ QR so that $\langle g \rangle =$ QR and $m \in$ QR.

Fix II. Choose almost regular hash function $h : G \to \{0,1\}^\ell$ and define $\mathcal{E}_{\mathsf{pk}}(m) = (g^k, h(g^{xk}) \oplus m)$ for $m \in \{0,1\}^\ell$. Then $h(g^{xk})$ is almost uniform.

Hybrid encryption

Assume that $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ is a IND-CPA secure cryptosystem and prg is a secure pseudorandom generator (secure stream-cipher, e.g. AES in counter mode).

Encrypt. For $m \in \{0,1\}^{\ell}$ choose seed $\in \mathcal{M}$ randomly and compute

$$\mathcal{E}^*_{\mathsf{pk}}(m) = (\mathcal{E}_{\mathsf{pk}}(\mathsf{seed}), \mathsf{prg}(\mathsf{seed}) \oplus m)$$

Decrypt. Given (c_1, c_2) compute seed $\leftarrow \mathcal{D}_{\mathsf{sk}}(c_1)$ and output $c_2 \oplus \mathsf{prg}(\mathsf{seed})$.

Theorem. The hybrid encryption is IND-CPA secure.

All homomorphic encryptions are vulnerable

A cryptosystem is homomorphic if $\mathcal{E}_{\mathsf{pk}}(m_1) \cdot \mathcal{E}_{\mathsf{pk}}(m_2) = \mathcal{E}_{\mathsf{pk}}(m_1 \circ m_2)$.

- Vanilla RSA is homomorphic.
- ElGamal is homomorphic.
- Goldwasser-Micali is homomorphic.

If Malice can somehow decrypt limited number of messages
$\Longrightarrow$ he can perfectly hide what messages are actually decrypted.

Sometimes decryption of few carefully selected cryptograms may leak enough information so that Malice can completely break the scheme.