A Taxonomy of Encryption-Scheme Security

Moni Naor
Weizmann Institute

Sources/Credits

- Cramer & Shoup, *A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack*.
- Dolev, Dwork & Naor, *Non-malleable Cryptography*.
- Dwork & Naor, *Method for message authentication from non-malleable crypto systems* (Patent).
- Dwork, Naor & Sahai, *Concurrent Zero-Knowledge*.

Rigorous Specification of Security

To define security of a system must specify:

1. The power of the adversary - both computational *and* access to the system.

2. What constitute a failure of the system - what it means to break the system.

Hardness assumption: no machine with limited resources can solve instances of a problem. More specifically:

- Distribution on instances,
- Prob. of success.

A security proof of a system *implementation*: no adversary with limitation as (1) can break the system as in (2), assuming the hardness assumption.

Taxonomy of Signature-Schemes

GMRiv classified according to:

Attacks

- Key-only attacks
- Generic chosen message attack
- Non-Adaptive chosen message attack
- Adaptive chosen message attack

What it means to break the scheme

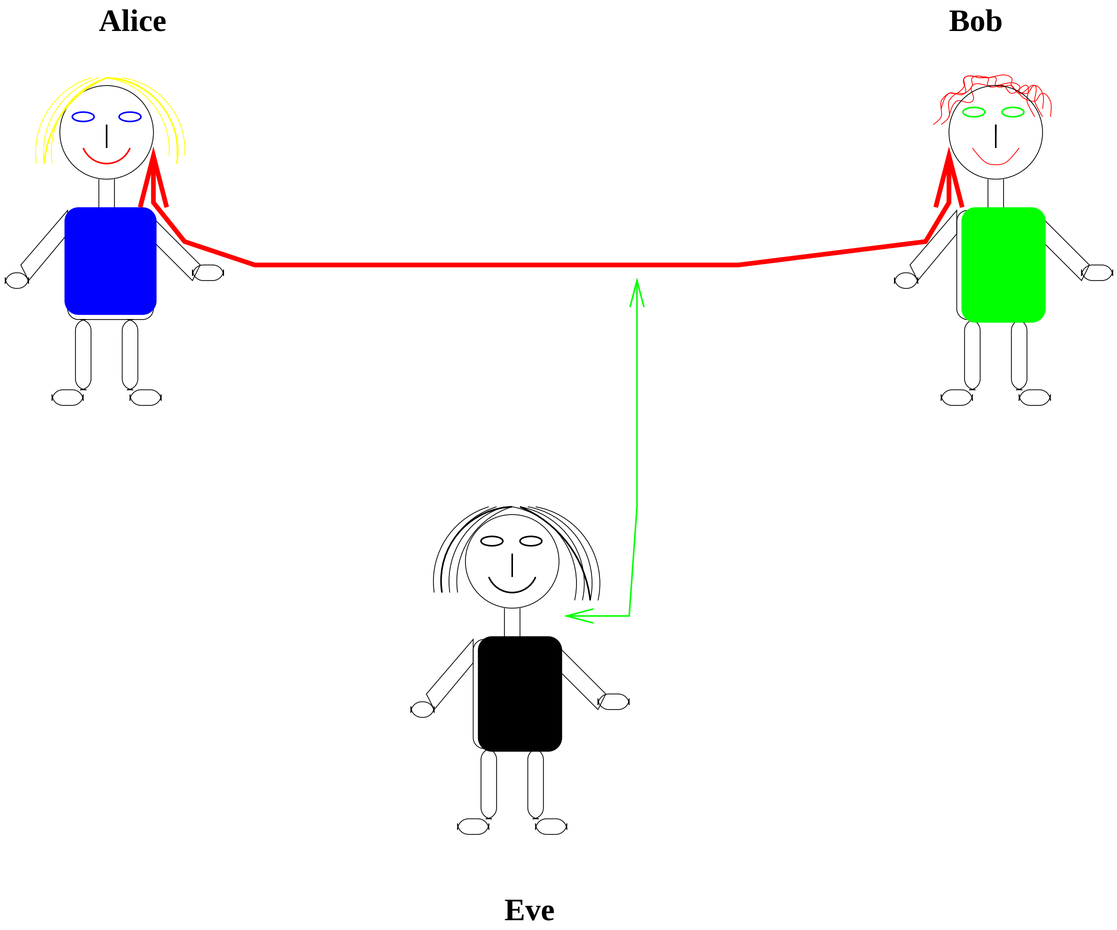
- Universal forgery ($\approx$ key-recovery)
- Selective forgery - message chosen a priori.
- Existential forgery - some message is forged.

All combination of attacks/breaking are relevant.

Strongest type: *Existential unforgeability against and adaptive chosen message attack.*

Encryption

Traditionally: Alice and Bob want to talk while Eve listens.

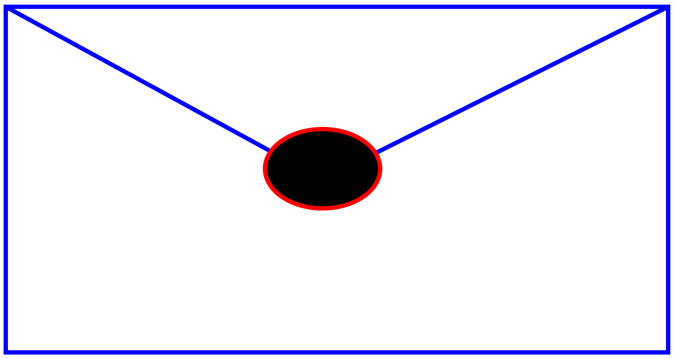


Public-key Cryptography - separate encryption from decryption keys:

- Alice "publishes" her public-key K_A
- Anyone who knows K_A key can send Alice a message so that only she should be able to decrypt it.

Sealed Envelopes

Intuition for security - sealed envelopes.



Try and mimic properties of sealed envelopes.

However - analogy cannot be perfect:

- Adversary can always duplicate an encrypted (digital) message.

- Adversary can always partition sealed envelope iand put parts in *other* sealed envelopes.

Therefore must pick the most desired properties.

Encryption - Attacks

- Chosen Plaintext

  Minimal attack relevant to PKCs. Must assumes adversary has no access to (any) decrypted messages!

- Chosen Ciphertext - preprocessing mode (Lunch-break). Challenge ciphertext is given *after* adversary relinquishes control of decryption device.

  Proposed by NY. May be sufficient to end-to-end. Good model for membership queries in computational learning.

- Chosen Ciphertext - postprocessing mode. Challenge ciphertext is *known* when the attacks takes place (but cannot submit it...).

  Proposed by Rackoff & Simon. Important in many protocols.

More options in private-key world...

Encryption - Notions of (Non)-Breaking

Semantic Security - whatever is computable about the plaintext *given* the ciphertext is computable *without* it.

Minimal notion of security for *single* encrypter.

Defined by Goldwasser and Micali.

Non-malleable security - whatever is computable in an encrypted form about the plaintext *given* the ciphertext is computable *without* it.

Important for achieving independence of messages.

Defined by DDN.

Semantic Security - Formalization

E - an Encryption scheme.

Adversary $\mathcal{A}$ receives challenge ciphertext $c \in_R E(m)$ where $m \in_R \mathcal{M}$ and produces x.

Competing $\mathcal{A}'$ does *not* receive $E(m)$ and produces x'.

Def 1 E is semantically secure if for every PPT relation $R(m, x)$ and all PPT adversaries $\mathcal{A}$ there exists a PPT $\mathcal{A}'$ such that

$$|\pi(\mathcal{A}, R) - \pi'(\mathcal{A}', R)|$$

is subpolynomial, where

$\pi(\mathcal{A}, R) = \Pr[x \text{ satisfies } R(m, \cdot)]$ and
$\pi'(\mathcal{A}', R) = \Pr[x' \text{ satisfies } R(m, \cdot)]$.

The outputs of $\mathcal{A}$ and $\mathcal{A}'$ are indistinguishable!

NM Security - Formalization

E - an Encryption scheme.

Adversary $\mathcal{A}$ receives challenge ciphertext $c \in_R E(m)$ where $m \in_R \mathcal{M}$ and produces x.

Competing $\mathcal{A}'$ does *not* receive $E(m)$ and produces x'.

Def 2 E is non-malleable if for all PPT relations $R(x, m)$ and all PPT $\mathcal{A}$ there exists a PPT $\mathcal{A}'$ such that

$$|\pi(\mathcal{A}, R) - \pi'(\mathcal{A}', R)|$$

is subpolynomial, where

$\pi(\mathcal{A}, R) =$ Pr[y satisfies $R(m, \cdot)$] and
$\pi'(\mathcal{A}', R) =$ Pr[y' satisfies $R(m, \cdot)$] and
y is the decryption (or decryptions) of x and
y' is the decryption (or decryptions) of x'.

The decryption of the outputs of $\mathcal{A}$ and $\mathcal{A}'$ are indistinguishable!

Who chooses $\mathcal{M}$? The adversary (or its simulator).

Example: Auctions

Different requirements - different notions.

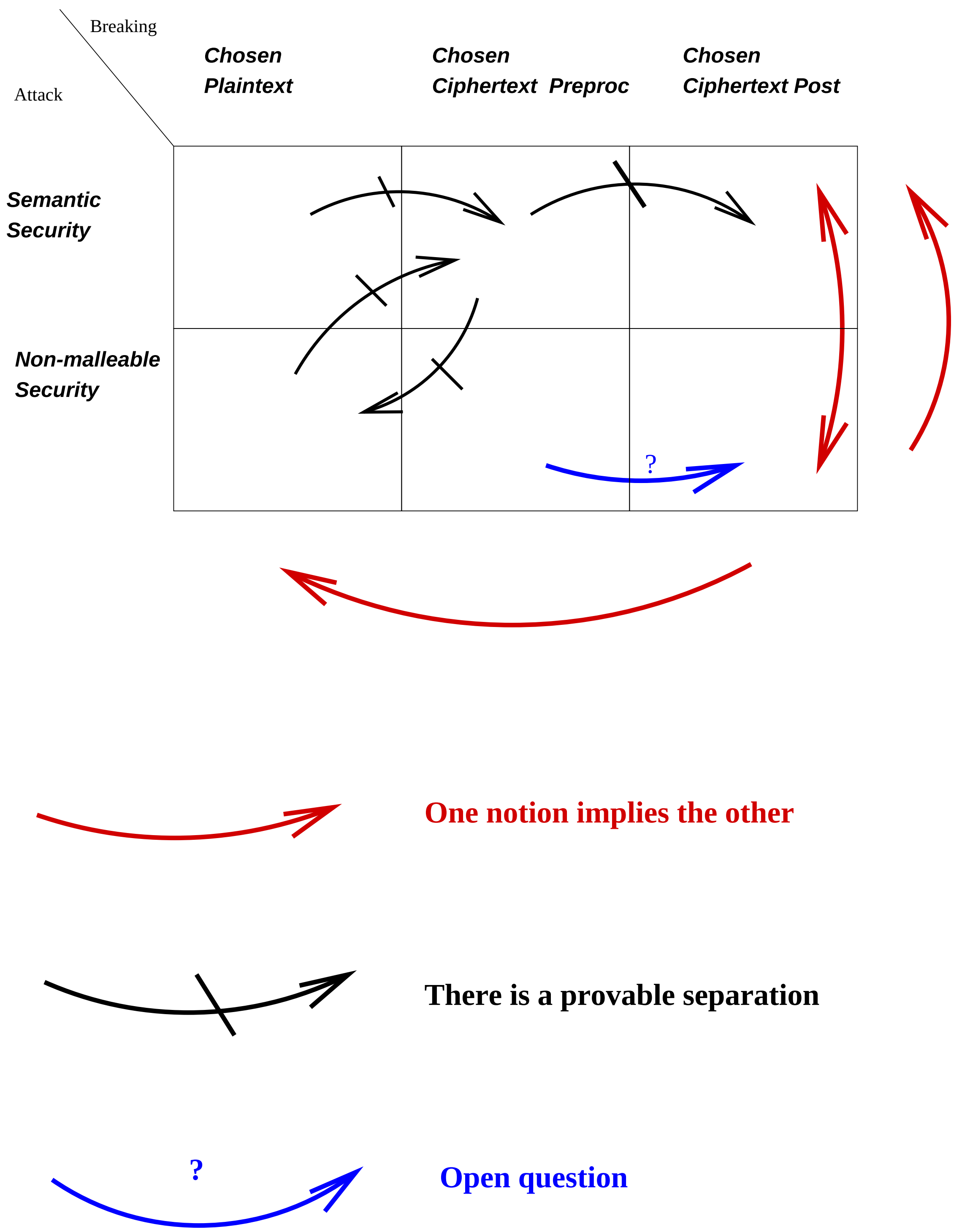
Semantic security is *not* sufficient for guaranteeing the independence of bids.

If key is used for a Single auction and secrecy is not required after the auction is over - **Non-malleable security against chosen plaintext attacks**.

If key is used for many auctions and secrecy is not required after the auction is over - **Non-malleable security against chosen ciphertext attack in the preprocessing mode**.

If key is used for many auctions and secrecy is required after the auction is over - **Non-malleable security against chosen ciphertext attack in the postproc. mode**.

The Possible Combinations



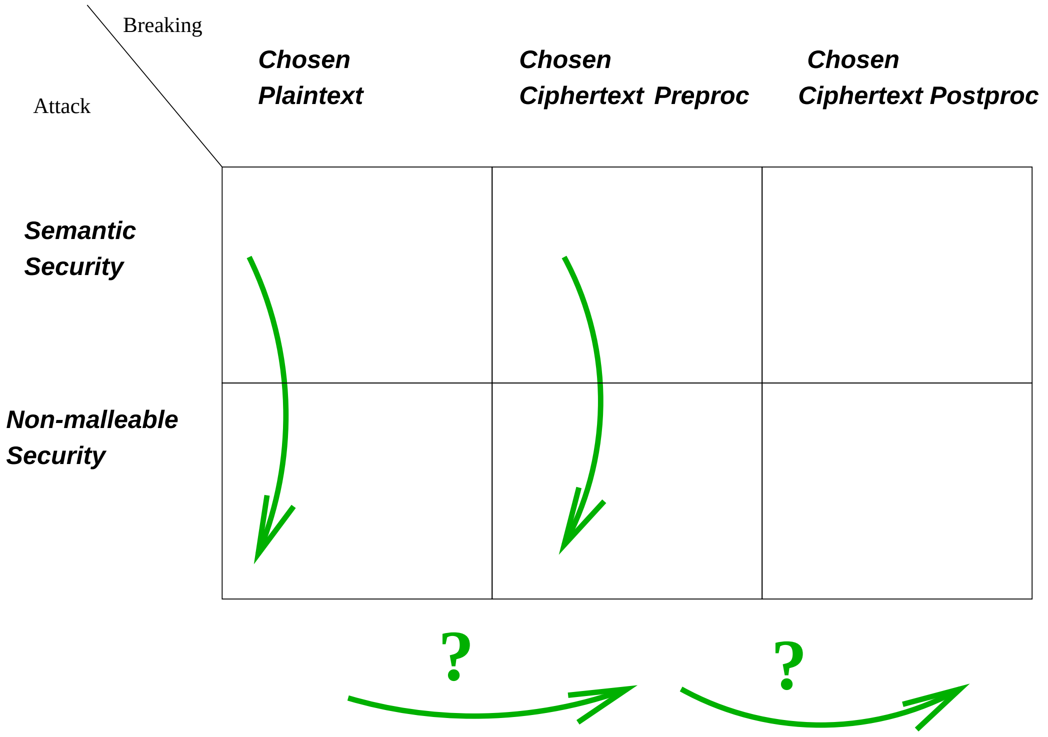
Examples

Breaking / Attack	*Chosen Plaintext*	*Chosen Ciphrtext Preproc*	*Chosen Ciphrtext Post*
Semantic Security	Rabin GM	Cramer-Shoup Lite	
Non-malleable Security	DDN Lite		DDN Cramer-Shoup

Applications

Breaking / Attack	*Chosen Plaintext*	*Chosen Ciphertext Preproc*	*Chosen Ciphertext Postproc*
Semantic Security	Single Encrypter Confidentiality of Decryptions	Membership Learning Simultaneous Decryptions	E-mail Authentication Protocols
Non-malleable Security	One-Time Auction	Many-Time Auction	Auction + Secrecy

Constructibility



Can construct one from the other

?

Not known how to construct one from the other

...Constructibility

If trapdoor permutations exist then PKCs exist that are

- Semantic security against chosen plaintext attacks - Using the GL hard-core bit.
- NM-secure against CCA in the posrproc. mode - using NIZK [DDN] (rather inefficient).

Cramer & Shoup: Under the *Decisional* DH assumption there is an *efficient* construction of NM secure against a CCA in the postproc. mode.

However, from trapdoor predicates we only know how protect (NM) against chosen plaintext attacks.

Examples:

- Ajtai-Dwork Cryptosystem.
- *Computational* DH Assumption.

(Modified) Rabin

Public-Key: $N = P \cdot Q$ such that $P, Q \equiv 3 \bmod 4$.

Private-Key: Factorization of N: P and Q.

Encryption of a message $m \in \{0, 1\}$: choose random $x \bmod N$ such that lsb of x is m and x is a quadratic residue $\bmod N$. Send $x^2 \bmod N$.

Decryption of a message y - find the square-root of y that is a q.r. $\bmod N$. Output its lsb.

Properties: The scheme is semantically secure against chosen plaintext attacks. It is *provably* insecure against chosen ciphertext attacks.

Goldwasser-Micali

Public-Key: $N = P \cdot Q$ such that $P, Q \equiv 3 \bmod 4$ and y, a quadratic non-residue $\bmod N$.

Private-Key: Factorization of N - P and Q.

Encryption of a message $m \in \{0, 1\}$: choose random $x \bmod N$ and output $y^m \cdot x^2 \bmod N$.

Decryption of a message z - if z is a q.r. output 0. Ow output 1.

Properties: The scheme is semantically secure against chosen plaintext attacks. It is *provably* malleable - to flip a bit multiply by y.

Principles for Increasing Security

Essentially all constructions achieving better than semantic security against chosen plaintext attacks use:

- Redundancy in the encryption.
- Validation that ciphertext is of the right form.

Validation is the trickiest part.

Relatively simple in

- Private-key Encryption
- Random Oracles

DDN Lite - The Idea

Obtaining non-malleability

- Start with a *semantically secure* PKC.
- Have many *independent* instances of the original scheme.
- Each encryption should use a *different* subset of the keys - enforce by one-time signatures.
- Before decryption - verify consistency.

Properties: If the original scheme is *Semantically Secure* against chosen plaintext attack the result is *NM-secure* against chosen plaintext attacks.

DDN Lite

Public-Key:

- $\langle K_1^0, K_1^1 \rangle, \langle K_2^0, K_2^1 \rangle, \ldots \langle K_n^0, K_n^1 \rangle$ where each K_i^b is the public-key of a semantic secure PKC.
- A function $h : \{0,1\}^* \mapsto \{0,1\}^n$ - UOHWF (hard to find collision with target input).

Private-Key: Decryption keys of $\{K_i^b\}$.

Encryption of a message m:

- Choose at random (K, L), the (public,private) keys of a *one-time signature scheme*.
- Let $b_1, b_2, \ldots b_n = h(K)$.
- Encrypt m using keys $K_i^{b_i}$ to obtain $C_1, C_2, \ldots C_n$.
- Sign $C_1, C_2, \ldots C_n$ using L and h. S, the result.
- the Ciphertext is $K, C_1, \ldots, C_n, S$.

Decryption of ciphertext $K, C_1, \ldots C_n, S$

- Verify the signature S on $C_1, C_2, \ldots, C_n$.
- Verify the equality of all the plaintexts.
- Decrypt using any one of the keys.

Cramer-Shoup Lite

Let $Q|P-1$ where P an Q are primes. Let g_1 and g_2 be generators of a subgroup of order Q mod P.

Private-Key: $x_1, x_2, z_1, z_2 \in_R Z_Q$

Public-Key: $c = g_1^{x_1} \cdot g_2^{x_2}$ and $h = g_1^{z_1} \cdot g_2^{z_2}$

Encryption of a message m (in the subgroup):

- Choose $r \in_R Z_Q$
- $u_1 = g_1^r$, $u_2 = g_2^r$, $e = h^r \cdot m$ and $v = c^r$.

Decryption of (u_1, u_2, e, v):

- verify that $v = u_1^{x_1} u_2^{x_2}$.
- If yes, $m = e/(u_1^{z_1} \cdot u_2^{z_2})$ mod P.

Properties: Semantically secure against CCA in the preproc. mode, but malleable.

Cramer-Shoup (the real thing)

Private-Key: $x_1, x_2, y_1, y_2, z_1, z_2 \in_R Z_Q$

Public-Key: $c = g_1^{x_1} \cdot g_2^{x_2}$, $d = g_1^{y_1} \cdot g_2^{y_2}$, $h = g_1^{z_1} \cdot g_2^{z_2}$ and H a UOWHF.

Encryption of a message m (in the subgroup):

- Choose $r \in_R Z_Q$
- Compute $u_1 = g_1^r$, $u_2 = g_2^r$, $e = h^r \cdot m$.
- Let $\alpha = H(u_1, u_2, e)$ and $v = c^r \cdot (d^r)^\alpha$.
- Ciphertext is (u_1, u_2, e, v).

Decryption of (u_1, u_2, e, v):

- verify that $v = u_1^{x_1} u_2^{x_2} \cdot (u_1^{y_1} u_2^{y_2})^\alpha$.
- If yes, $m = e/(u_1^{z_1} \cdot u_2^{z_2})$ mod P.

Properties: NM-secure against CCA in the postproc. mode.

Authentication Protocols

Public-key authentication - similar security requirements as signature schemes, but:

- Allow interaction
- Do not *require* third party verifiability

Public Key Authentication Protocol

P has a public key K of an encryption scheme E.

- $V \longrightarrow P : \gamma \in_R E_K(m \circ r)$, $r \in_R \{0,1\}^n$
- $P \longrightarrow V$: Verify that the prefix of the plaintext is the message. If yes, send r.

The scheme allows some sort of chosen ciphertext attack on E_K.

Sensitive to malleability: if given $E(m \circ r)$ can generate $E(m' \circ r)$ - can forge messages.

If E is NM-secure against CCA in the postproc. mode, the scheme is existentially unforgeable against an adaptive chosen message attack.

Deniable Authentication

What receipt does V obtain? Not much...

To claim deniability - need similar property to zero-knowledge.

Sufficient if V proves knowledge of r.

When authentication is performed concurrently - need (α, β) timing assumptions [DNS]

Using Non-Malleable PKC

1. $P \longrightarrow V : C(r_1)$
2. $V \longrightarrow P : r_2,\ E(r_2 \circ C(r_1) \circ r_3)$
3. $P \longrightarrow V : auth_{r_1 \oplus r_2}(m)$
4. $P \longrightarrow V : r_3,\ r_1$

Timing Constraints:

- P delays Step 4 until more than β time has elapsed on P's local clock since r_2 is received in Step 2.
- V accepts only if Step 3 is received within time α on V's local clock from Step 2.

Private-key World

Preventing CCA in the postproc. mode - relatively simple: add private-key authentication.

Shared key: s_1 and s_2 seeds to a pseudo-random function F.

To encrypt m:

- Choose random r
- Let $y = F_{s_1}(r) \oplus m$
- Let $z = F_{s_2}(r \circ y)$
- Send (r, y, z)

Properties: NM-secure against CCA in the postproc. mode. No adversary can find another ciphertext.

Can combine with public-key for efficiency!

Weaker attacks:

- Permutations (Block-ciphers)
- Static attacks.
- Remotely-keyed Encryption.