Detecting Dangerous Queries:

A New Approach for Chosen Ciphertext Security

Susan Hohenberger

Allison Lewko

Brent Waters

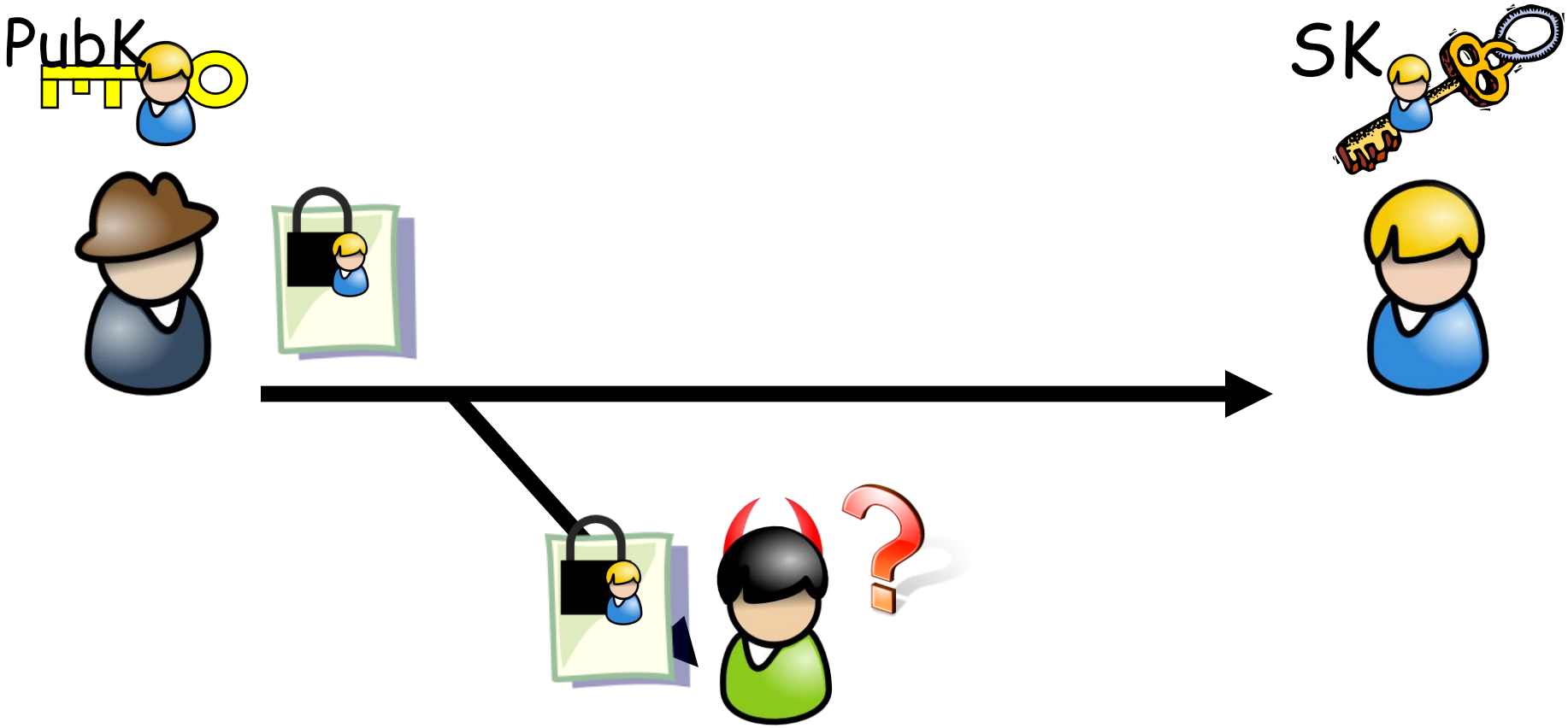
JOHNS HOPKINS UNIVERSITY

THE UNIVERSITY OF TEXAS AT AUSTIN

THE UNIVERSITY OF TEXAS AT AUSTIN

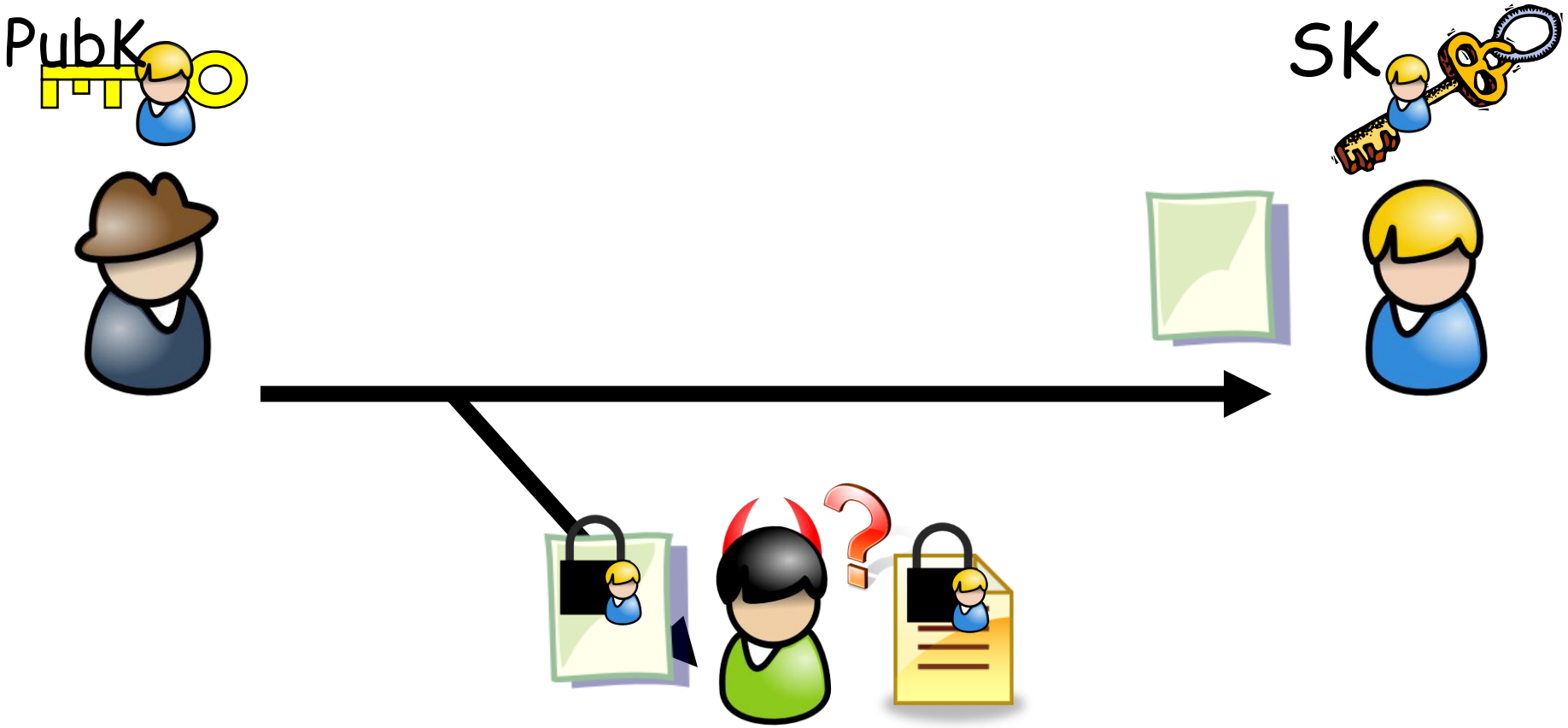
Public Key Encryption [DH76,RSA78,GM84]

Passive Attacker : Chosen Plaintext Attack (CPA)



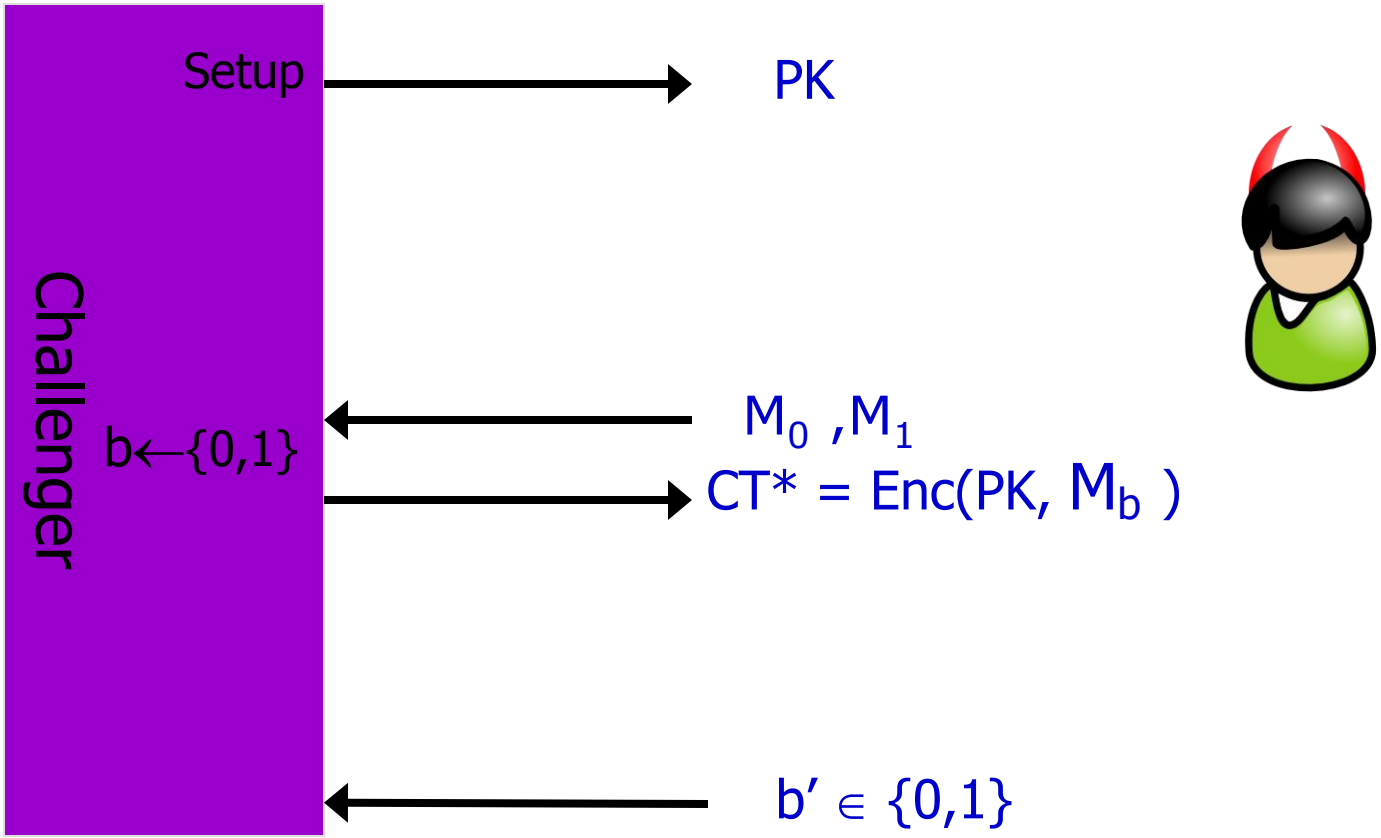
Active Attackers [NY90,DDN91,RS91]

Chosen Ciphertext Attack (CCA)



IND-CPA [GM84]

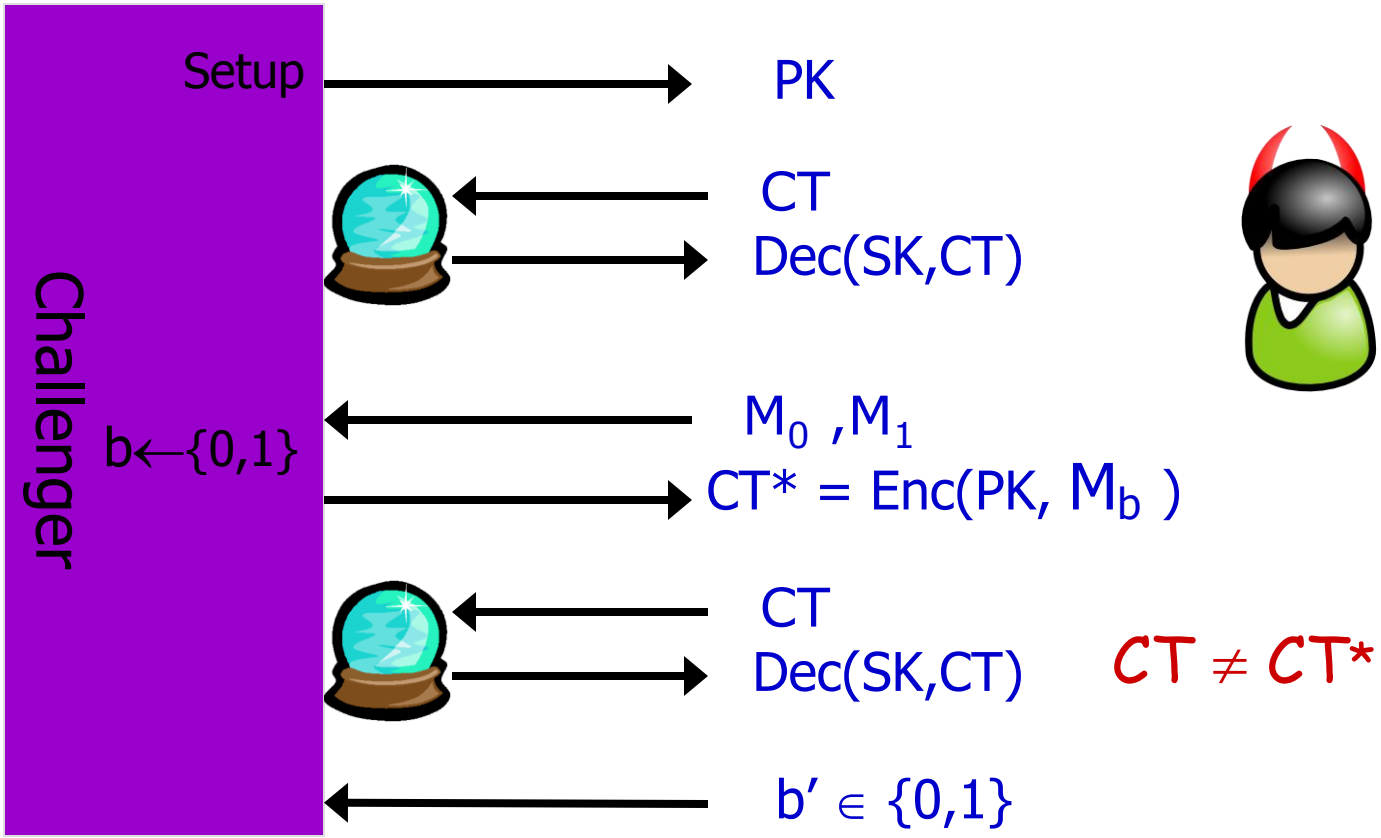
Indistinguishability under Chosen Plaintext Attack



$Adv_A = Pr[b=b']-1/2$

IND-CCA [NY90,DDN91,RS91]

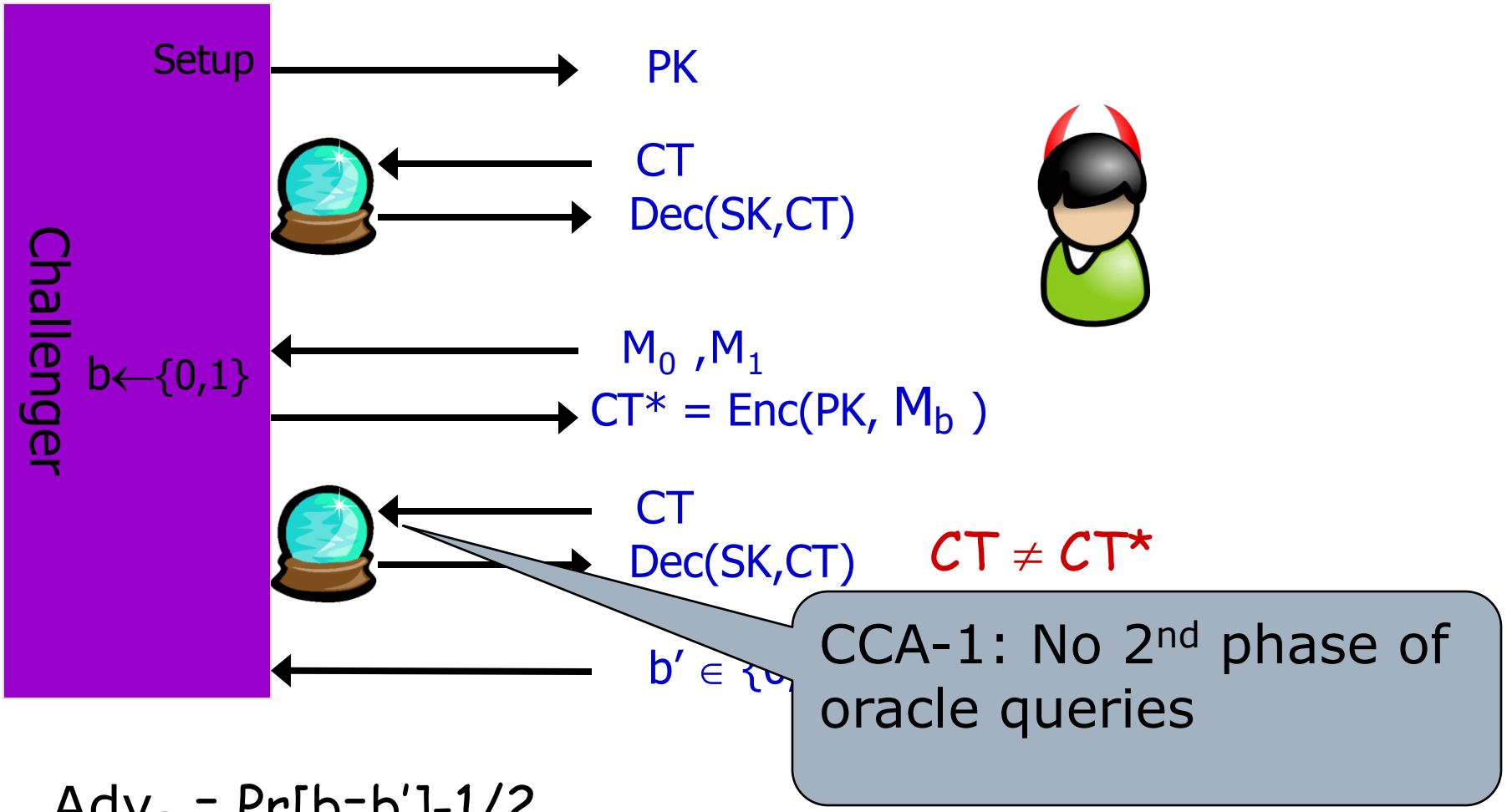
Indistinguishability under Chosen **Ciphertext** Attack



$Adv_A = Pr[b=b']-1/2$

IND-CCA [NY90,DDN91,RS91]

Indistinguishability under Chosen **Ciphertext** Attack



$Adv_A = Pr[b=b']-1/2$

The Grand Goal: CCA from CPA

CCA

CPA

Some Prior Methods (Standard Model)

NIZK [BFM88,NY90,DDN91,RS91,S99]

- TPD/RSA, Pairings No:DDH, Lattices

Cramer-Shoup plus [CS98,02,…]

- DDH,DCR, Factoring, IBE [CHK04], No:Lattices

Lossy TDFs [PW08,RS09,…]

- DDH, Lattices

1-bit CCA to n-bit CCA [MS09]

- Straightforward appending won't work!

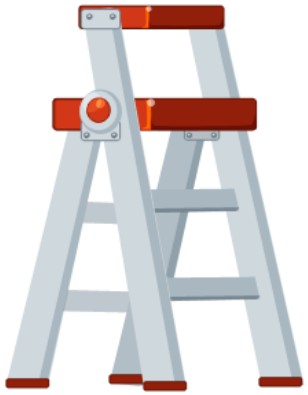
- Neat ideas
- Heavyweight machinery + complex
- We will adapt + generalize some ideas

Our Result

New General Approach for CCA security:

Detectable Chosen Ciphertext Security (DCCA)

CCA



DCCA

DCCA Security: Intuition

CCA secure *if* avoid "dangerous" queries

1) Hard to produce bad queries w/o challenge CT

2) Can detect dangerous queries

Example: Concatenate 1 bit CCA ciphertexts

CT* 

Dangerous Query for CT*: CT = Reorder of CT*

1)Hard to produce w/o CT* 2) Easy to detect

Detectable Encryption System

Setup(1^n) $\rightarrow$ (PK,SK)

Encrypt(PK,M) $\rightarrow$ CT

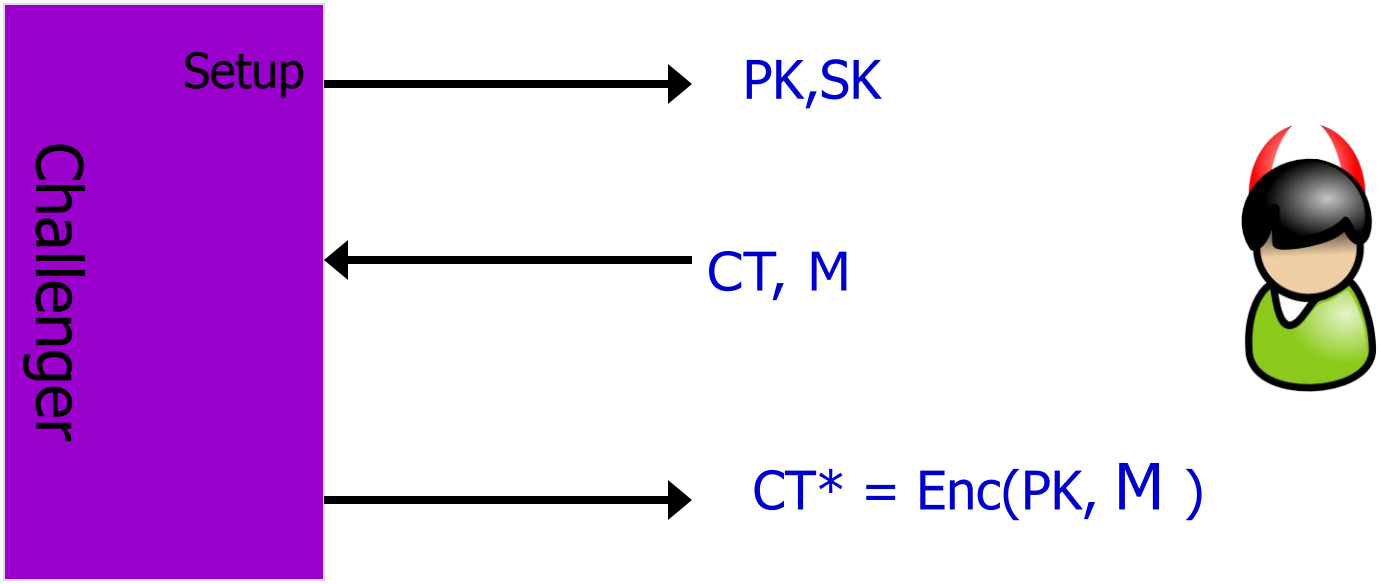
Decrypt(SK,CT) $\rightarrow$ M

F(PK, CT* , CT) $\rightarrow$ {0,1}

Outputs '1' if CT is a "dangerous" query for CT*

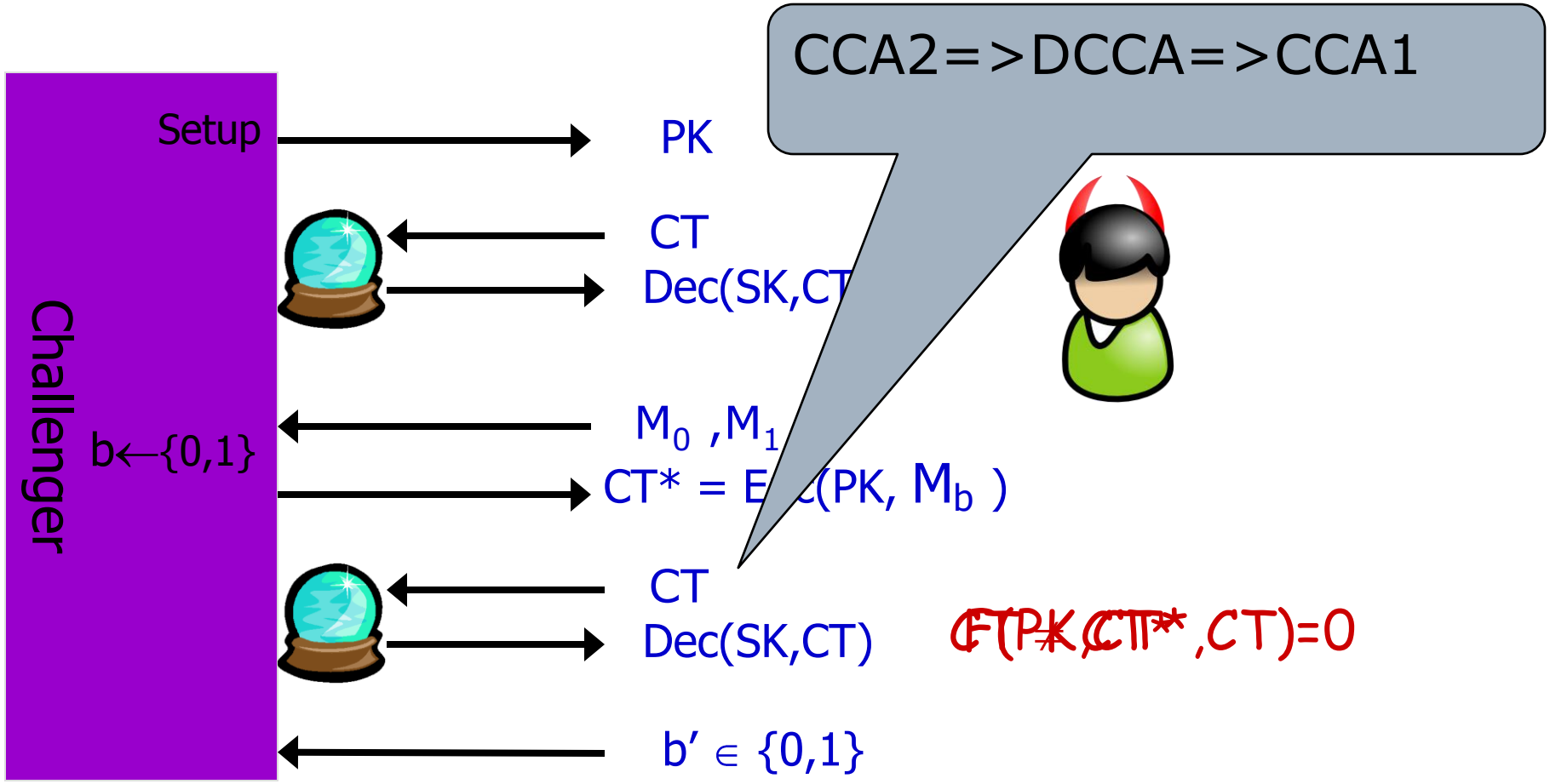
Two Security Properties

Property 1: Hard to Predict (Strong)



$Adv_A = Pr[F(PK,CT,CT^*)=1]$

Property 2: Indistinguishability



$Adv_A = Pr[b=b']-1/2$

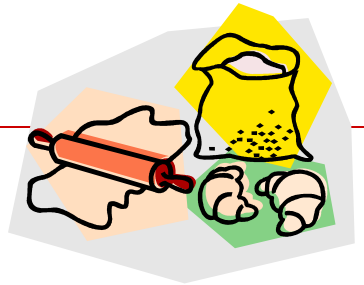
Examples

One bit to many bit CCA

Tag-Based Encryption [MRY04,K06]

Sloppy/Heuristic CCA

The Ingredients



Msg $\in \{0,1\}^*$ and randomness $\in \{0,1\}^n$
Justified by Pseudo Random Generators

1-Bound

Trivial

Detectable CCA

Our Construction

Setup

Setup(1^n):

1) $\text{Setup}_{1B}(1^n) \rightarrow (PK_A, SK_A)$
2) $\text{Setup}_{CPA}(1^n) \rightarrow (PK_B, SK_B)$
3) $\text{Setup}_{DCCA}(1^n) \rightarrow (PK_{in}, SK_{in})$

$PK = PK_A, PK_B, PK_{in}$

$SK = SK_A, SK_B, SK_{in}$

Encryption

Encrypt(PK,M):

1) Choose random r_a ,r_b , $r_{in} \in \{0,1\}^n$
2) $C_{in} = Enc_{DCCA}(PK_{in}, (M,r_a, r_b) ; r_{in})$
3) $C_A=Enc_{1B} (PK_A, C_{in}; r_a)$, $C_B=Enc_{CPA} (PK_B, C_{in}; r_b)$
4) $CT= C_A , C_B$

Decryption

Decrypt(SK, $CT = (C_A , C_B)$) :

1) $C_{in}' = Dec(SK_A , C_A)$

2) $(M', r_a', r_b') = Dec(SK_{in} , C_{in}')$

3) $C_A' = Enc_{1B} (C_{in}'; r_a')$, $C_B' = Enc_{CPA} (C_{in} ; r_b')$

4) If $C_A \neq C_A'$ OR $C_B \neq C_B'$ reject ;else M'

$C_A =$ $(M, r_a , r_b); r_{in}$ $; r_a$ $C_B =$ $(M, r_a , r_b); r_{in}$ $; r_b$

Idea: Recover (M, r_a , r_b) then re-encrypt

A Few Comments

$C_A = [\,(M, r_a, r_b); r_{in}\,; r_a]$ $C_B = [\,(M, r_a, r_b); r_{in}\,; r_b]$

Features: Naor-Yung 2-key & Myers-shelat nesting

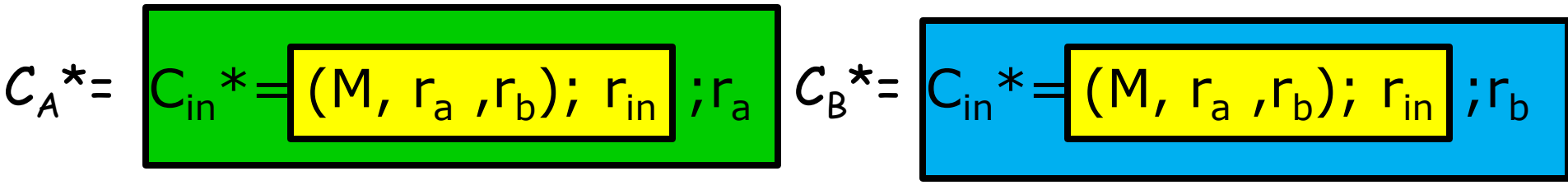
Embedded Randomness vs. NIZK

Proof w/ embedding randomness:

Good: Decrypt from either side

Problem: Embedding challenge

What is the trouble?



Challenge CT= C_A *, C_B * encryptions of C_{in} *

Problem Query: Get C_{in}' s.t. $F(PK_{DCCA}, C_{in}*, C_{in}') = 1$

Bad Event: Query C= C_A , C_B s.t.

(1) $C_A \neq C_A$ *

(2) Dec(SK_A, C_A) = C_{in}' where $F(PK_{DCCA}, C_{in}*, C_{in}') = 1$

Nested Indist. Game

Attacker gets

If prove under this game we are done!

Challenge Inner en… Msg + randomness or all 0's

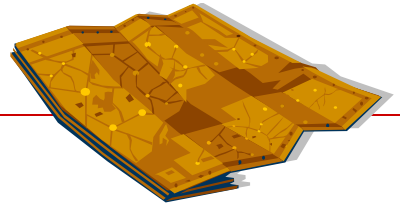
z=1

$C_A^* = C_{in}^* = (M, r_a, r_b); r_{in}; r_a$ $C_B^* = C_{in}^* = (M, r_a, r_b); r_{in}; r_b$

z=0 No embedded randomness

$C_A^* = C_{in}^* = (00\ldots00); r_{in}; r_a$ $C_B^* = C_{in}^* = (00\ldots00); r_{in}; r_b$

Proof Overview



Eliminate bad event => Security follows from DCCA

(1) Eliminate with z=0 (no embedded randomness)

(2) Indirectly infer z=1 case from (1)

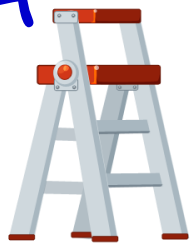
(3) Finish off

Summary

- New abstraction: Detectable *CCA* security
- Build *CCA* from it
- Cover 1 to many bit enc. , tag-based, & more
- Embedded randomness --- blessing & problems
- Indirect inference on bad event

Our Picture (not necessarily to scale)

CCA



DCCA

CCA-1

CPA

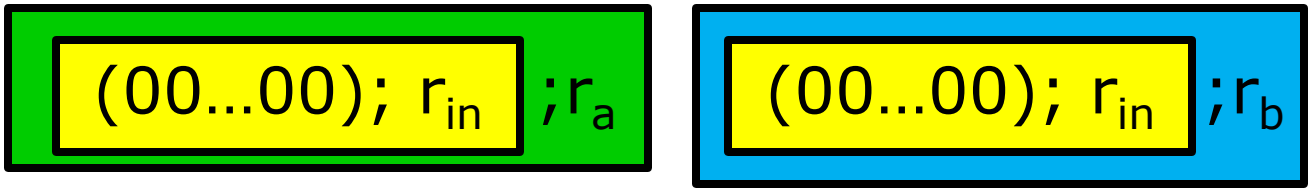
Thank you

Bad Event Analysis (no embedded randomness)

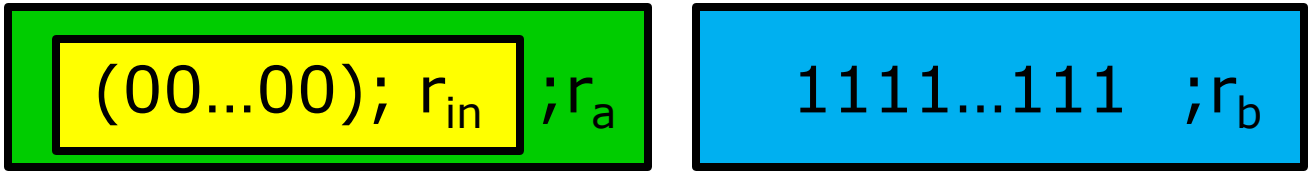
Show probabilities are close

Nested

IND-CPA



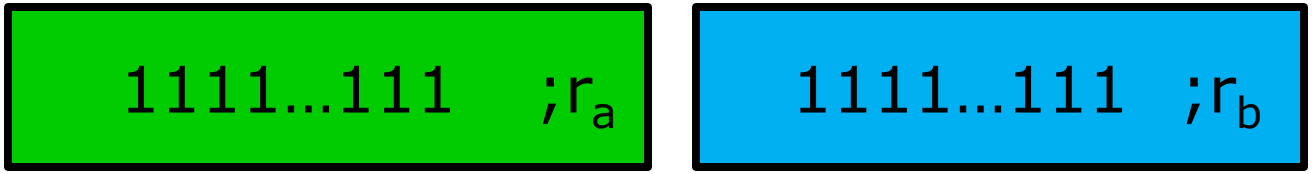
Right-Erased



Switch -Decrypt

1Bounded CCA

Full-Erased



=negl(n) unpredictability

No Bad Event for embedded randomness

Suppose it did happen => We break DCCA indist.

1) Run Indist Game on A (while playing DCCA)

2) Submit $Msg_1 = (M, r_a, r_b)$, $Msg_0 = (00…00)$

3) Get back either $(M, r_a, r_b); r_{in}$ or $(00…00); r_{in}$

4) Create challenge CT (know SK_A, SK_B)

5) Use DCCA oracle to answer non-dangerous queries

What if get dangerous query? Stuck!

But then we know it must be Msg_1 => breaks DCCA!

Finishing it off

z=1

$C_A^* = C_{in}^* = (M, r_a, r_b); r_{in}; r_a$ $C_B^* = C_{in}^* = (M, r_a, r_b); r_{in}; r_b$

z=0 No embedded randomness

$C_A^* = C_{in}^* = (00\ldots00); r_{in}; r_a$ $C_B^* = C_{in}^* = (00\ldots00); r_{in}; r_b$

N.I. easy to prove from DCCA if no bad events

CCA security follows immediately

Could CCA-1 work?

Idea: Replace DCCA component w/ CCA-1

Problem 1: Proof needs to detect

Problem 2: Counterexample (w/natural CCA-1 scheme)

Ex. 1: n-bit DCCA from 1 bit CCA

Idea: Use basic concatenation

Enc(PK,m) $\rightarrow$ C1=Enc(PK,m_1), …, Cn=Enc(PK,m_n)

F(PK,CT*,CT): $\exists$ (i,j) s.t. CT_i*=CT_j

Ex. 2: Tag-Based Encryption [MRY04,K06]

Tag-Based Encryption:

(1) Each ciphertext associated with a tag

(2) Is CCA secure as long as Tag_{CT^*} not queried

F(PK,CT*,CT): $Tag_{CT^*} = Tag_{CT}$

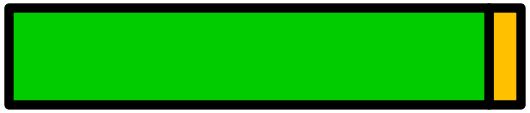
Examples: CHK04-lite, Kiltz06, PW08 (CCA-1 version), DDN91 (w/o signature)

Ex. 3: Heuristic/Sloppy CCA

Idea: DCCA easier to meet than CCA

(1) Heuristic approach

(2) Sloppy: E.g. "Slack" bit in group representation

CT: 

Apply transformation in case messed up

Could CCA-1 work?

Idea: Replace DCCA component w/ CCA-1

Problem 1: Proof needs to detect

Problem 2: Can create an oracle that breaks it

(CT*) :Decrypts CT*, encrypts M in another CT'

Q1: The oracle is strong! Is there middle ground?

Q2: Structure for CCA-1? Proof idea?

Prior Methods (Standard Model)

NIZK [BFM88,NY90,DDN91,RS91,S99]

- NIZK proves well formness
- NIZKs are rare: TPD/RSA, Pairings No:DDH, Lattices

Cramer-Shoup plus [CS98,02,…]

- Efficient systems from number theory
- DDH,DCR, Factoring, IBE [CHK04], No:Lattices

Prior Methods (Standard Model)

Lossy TDFs [PW08,RS09,…]

- Randomness recovery => use to verify CT
- Change PK in proof
- DDH, Lattices

1-bit to many bit *CCA*[MS09]

- General techniques
- Partial randomness recovery

BE-Nested vs. BE-Right-Erase

Standard IND-CPA reduction

- Know SK_A, SK_{in} , not SK_B
- Observe BE using SK_A

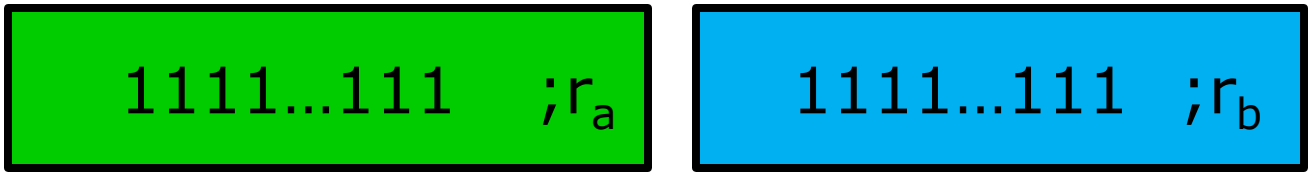
Switch Decrypt

Switch from using SK_A to SK_B to decrypt

- These are equivalent from Attacker's view
- Best of both worlds: Challenge CT not embed randomness, but queries must!

BE-Right-Erased vs. BE-Full-Erased

Full-Erased



C_{in}*= (00…00); r_{in} *is gone!*

Unpredictability: Pr[Bad event in Full Erase] = negl(n)

BE-Right-Erased vs. BE-Full-Erased

1-Bounded *CCA* reduction

- Know SK_B, SK_{in} , not SK_A
- Problem: Cannot observe bad event using SK_B
- Solution: "Peek" at 1 *A* query using 1-Bounded 1/Q chance of seeing it