Security Notions for Broadcast Encryption

D. Hieu Phan, David Pointcheval, *Mario Strefler*

ENS
strefler@di.ens.fr

2011 June 09

1 Motivation

2 Our security model

3 Fully adaptive security

4 Choice of the target set

5 A fully secure scheme

6 Conclusion

Broadcast Encryption

- N users $\{u_1, \ldots u_N\} = U$
- Here: Key encapsulation mechanism
- Goal: Encrypt K to any $S \subset U$
- Security definition? (Different in most papers)

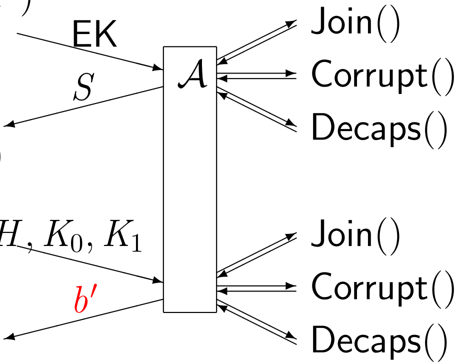
Security of BE

$(\mathsf{MSK}, \mathsf{EK}) \leftarrow \mathit{Setup}(1^k)$

$(H, K) \leftarrow \mathit{Enc}(\mathsf{EK}, S)$

$K_b \leftarrow K, K_{1-b} \xleftarrow{\$} \mathcal{K}$

win if $b = b'$

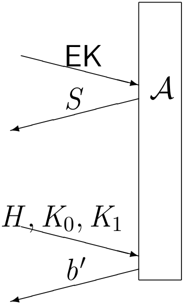


Restrictions:

- no corrupted users in S
- don't query decaps on H

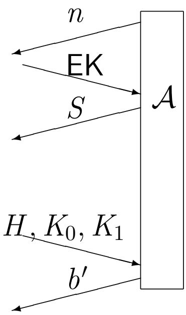
Security Notions

- Dynamic (join oracle)
- Adaptive corruption
- Decryption oracle
- Choice of the target set



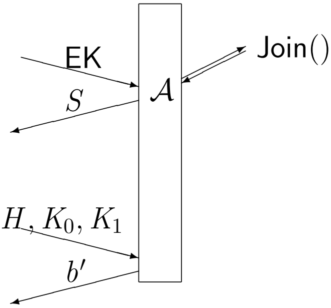
Security Notions

- Dynamic (join oracle)
  - static (fixed at setup)
- Adaptive corruption
- Decryption oracle
- Choice of the target set



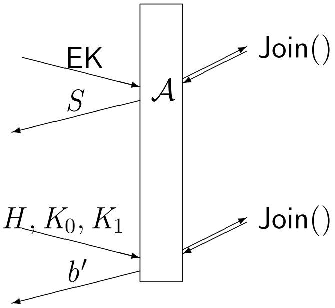
Security Notions

- Dynamic (join oracle)
  - static (fixed at setup)
  - dynamic1
- Adaptive corruption
- Decryption oracle
- Choice of the target set



Security Notions

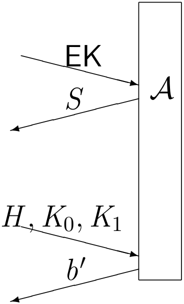
- Dynamic (join oracle)
    - static (fixed at setup)
    - dynamic1
    - dynamic2
- Adaptive corruption
- Decryption oracle
- Choice of the target set



Security Notions

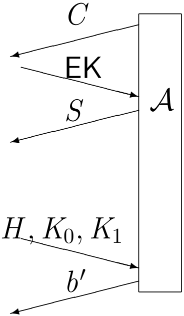
- Dynamic (join oracle)
- Adaptive corruption
  - no corruption

- Decryption oracle
- Choice of the target set



Security Notions

- Dynamic (join oracle)
- Adaptive corruption
  - no corruption
  - selective corruption

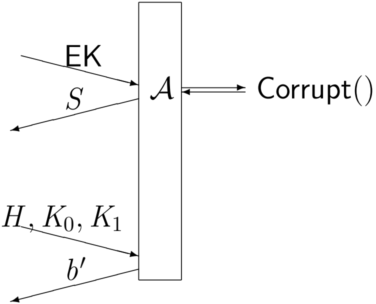


- Decryption oracle
- Choice of the target set

Security Notions

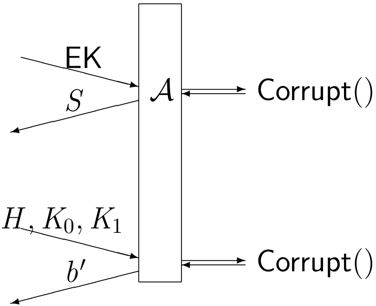
- Dynamic (join oracle)
- Adaptive corruption
    - no corruption
    - selective corruption
    - adaptive1

- Decryption oracle
- Choice of the target set



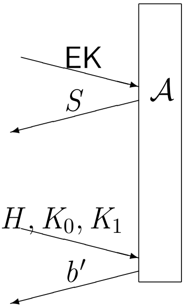
Security Notions

- Dynamic (join oracle)
- Adaptive corruption
  - no corruption
  - selective corruption
  - adaptive1
  - adaptive2
- Decryption oracle
- Choice of the target set



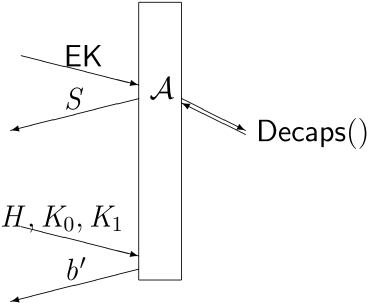
Security Notions

- Dynamic (join oracle)
- Adaptive corruption
- Decryption oracle
  - CPA
- Choice of the target set



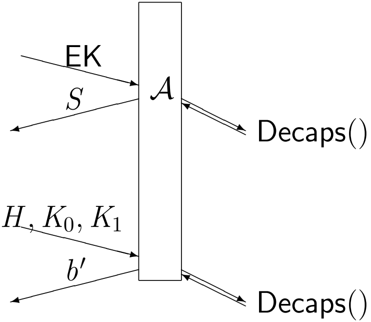
Security Notions

- Dynamic (join oracle)
- Adaptive corruption
- Decryption oracle
  - CPA
  - CCA1
- Choice of the target set



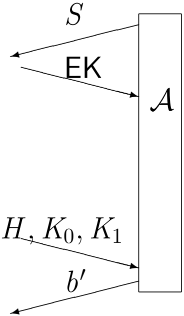
Security Notions

- Dynamic (join oracle)
- Adaptive corruption
- Decryption oracle
  - CPA
  - CCA1
  - CCA2
- Choice of the target set



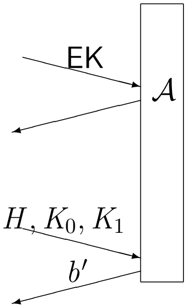
Security Notions

- Dynamic (join oracle)
- Adaptive corruption
- Decryption oracle
- Choice of the target set
    - chosen before setup



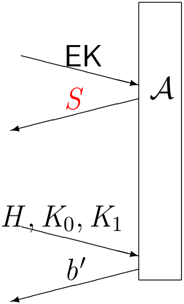
Security Notions

- Dynamic (join oracle)
- Adaptive corruption
- Decryption oracle
- Choice of the target set
  - chosen before setup
  - fixed to include all noncorrupted users



Security Notions

- Dynamic (join oracle)
- Adaptive corruption
- Decryption oracle
- Choice of the target set
  - chosen before setup
  - fixed to include all noncorrupted users
  - chosen by the adversary

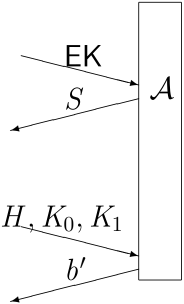


Security Notions

- Dynamic (join oracle)
- Adaptive corruption
- Decryption oracle
- Choice of the target set

Consider these independently

- Cannot corrupt users that don't exist
- Interactions between corruption and choice of target set



Adaptive Corruption

The security model of [GW09]:

- Setup: $(\mathsf{ek}, \mathsf{dk}) \leftarrow \mathsf{KeyGen}(1^k)$
- Give ek to $\mathcal{A}^{\mathsf{OCorrupt}(\cdot)}$
- Encrypt to adversarially chosen S

No second phase

Is there a difference? (as for CCA1 vs. CCA2)

Separating Adaptive1 from Adaptive2

- Only for t-collusion-resilient schemes, with t and $(N - t)$ non-constant
- Reason: $\binom{t}{N}$ exponential

Approach:

- Take an Ad$\mathbf{1}$-secure BE scheme Π
- Modify Π so it is clearly Ad$\mathbf{2}$-insecure, but remains Ad$\mathbf{1}$-secure

Separating Example

Π'.Encaps(EK, S):

$(H', K) \leftarrow \Pi$.Encaps(EK, S);
Choose a random subset $I \subset U$, with $|I| = t$;
$\forall i \in I : (H_i, K_i) \leftarrow \Pi$.Encaps(EK, $\{i\}$)
Set $K_0 = K \bigoplus_{i \in I} K_i$;
`return`$(H', K_0, \{H_i\}_{i \in I}), K$.

Only for CPA and CCA1
Example for CCA2 is more complicated

Choice of the Target Set

Model in [DF03]: Target set is automatically the set of uncorrupted users

- Setup: $(\mathsf{ek}, \mathsf{dk}) \leftarrow \mathsf{KeyGen}(1^k)$
- Give ek to $\mathcal{A}^{\mathsf{OCorrupt}(\cdot)}$
- Encrypt to anybody but R

Is there a difference? (Restricts the adversary)

Separating modes of choosing S

Theorem

All the following implications are strict.
In a model with no corruption or selective corruption,
choice of the target set $\Rightarrow$ *fixed taget set.*
In a model with adaptive1 or adaptive2 corruption:

- *For fully collusion-resilient BE schemes,*
  choice of the target set $\Leftrightarrow$ *fixed taget set.*
- *If the adversary must leave two users uncorrupted,*
  choice of the target set $\Rightarrow$ *fixed taget set.*

Equivalence (choice $\Leftrightarrow$ fixed)

Assume a fully collusion-secure scheme.

$\Rightarrow$ If adversary can choose S, can set it to $U \setminus \mathcal{C}$.

$\Leftarrow$ Let $\mathcal{A}^{choice}$ be a successful adversary who can choose S. Then we construct $\mathcal{A}^{fixed}$ as follows:

- $\mathcal{A}^{fixed}$ faithfully forwards all queries.
- When $\mathcal{A}^{choice}$ outputs his challenge target set S, $\mathcal{A}^{fixed}$ corrupts users so that $U \setminus \mathcal{C} = S$, then asks for the challenge and forwards it to $\mathcal{A}^{choice}$.
- He forwards the guess bit b and wins with the same probability as $\mathcal{A}^{choice}$.

$\mathcal{A}^{fixed}$ corrupts more users, which could reduce the tightness of a security proof.

Separation (choice $\Rightarrow$ fixed)

If the adversary must leave two users uncorrupted:

- If not all users can be corrupted, proof fails
- In this case, $\mathcal{A}^{choice}$ can choose S with $|S| = 1$
- Separating example: Scheme with pathological behaviour if $|S| = 1$ (e.g. $K = 0$)

Fully secure naive scheme

Let $\mathcal{PKE}$ be an IND-CCA2 secure PKE scheme with key length κ, $\mathcal{MAC}$ a SUF-CMA MAC.

- Setup(1^k) MSK $\stackrel{\text{def}}{=} \emptyset$; EK $\stackrel{\text{def}}{=} \emptyset$; *Reg* $\stackrel{\text{def}}{=} \emptyset$
- Join(MSK, i) $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathcal{PKE}$.KeyGen$(1^k)$.
- Encaps(EK, S): $K, K_m \stackrel{\$}{\leftarrow} \{0,1\}^k$;
  $\forall i \in S : c_i \leftarrow \mathcal{PKE}.\mathsf{Enc}(\mathsf{pk}_i, K||K_m)$;
  $\sigma \leftarrow \mathcal{MAC}_{K_m}(c_1||\ldots||c_{|S|})$;
  $H \stackrel{\text{def}}{=} c_1||\ldots||c_{|S|}||\sigma$
- Decaps(sk_i, S, H): $K||K_m = \mathcal{PKE}.\mathsf{Dec}(\mathsf{sk}_i, c_i)$
  `if` $\mathcal{MAC}$.Verify$(K_m, \sigma, c_1||\ldots||c_{|S|})$ `return` K,
  `else return` $\perp$

Summary

We

- Defined a clean hierarchy of security notions
- Showed separations / equivalence between all notions
- Showed that schemes exist that fulfill the strongest notion