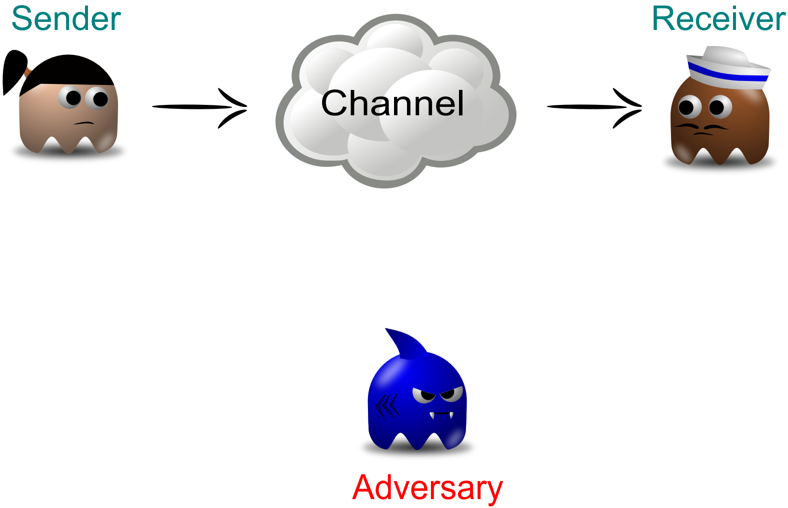
On Symmetric Encryption with Distinguishable Decryption Failures

Alexandra Boldyreva, **Jean Paul Degabriele**, Kenny Paterson, and Martijn Stam

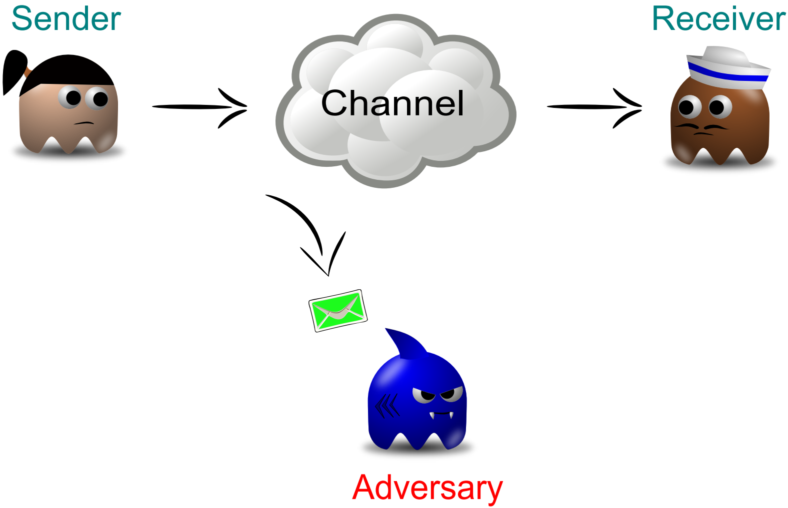
FSE - 12th Mar 2013

Outline

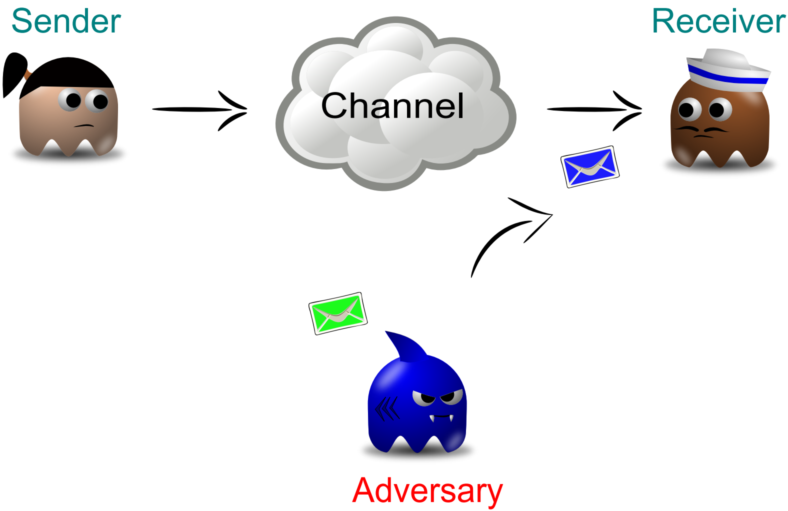
Attacks Based on Decryption Failures



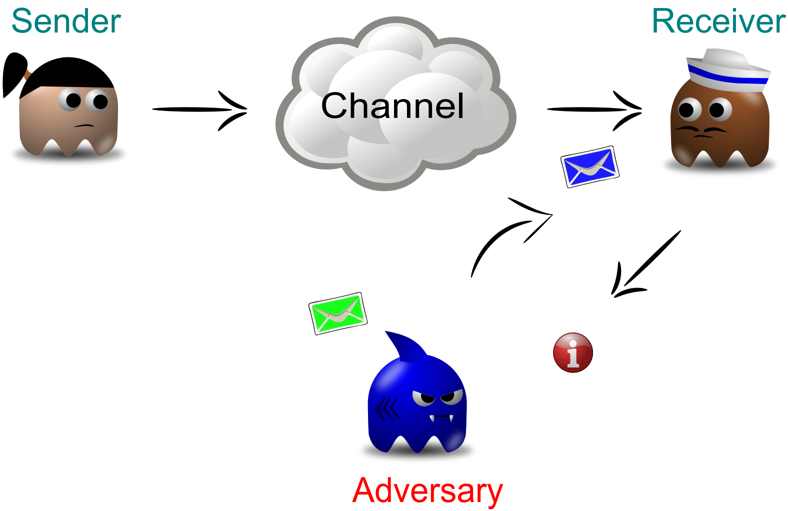
Attacks Based on Decryption Failures



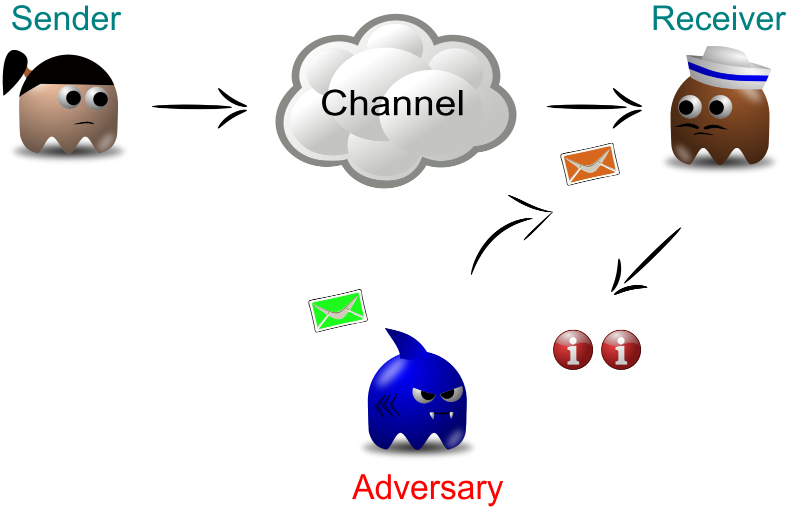
Attacks Based on Decryption Failures



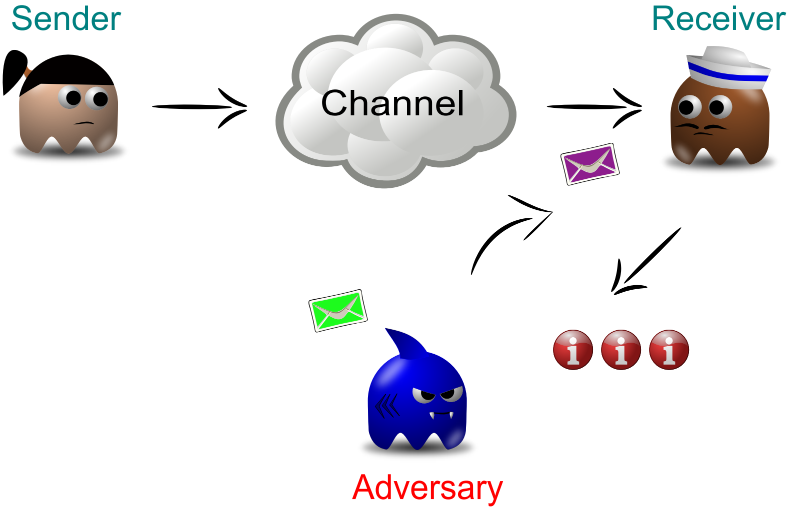
Attacks Based on Decryption Failures



Attacks Based on Decryption Failures



Attacks Based on Decryption Failures



Attacks Based on Decryption Failures

- The classic examples are Bleichenbacher's attack on RSA and Vaudenay's padding oracle attack on CBC encryption.

- These attacks motivated us to require IND-CCA security, but does IND-CCA always guard against such attacks?

Attacks Based on Decryption Failures

- The classic examples are Bleichenbacher's attack on RSA and Vaudenay's padding oracle attack on CBC encryption.
- These attacks motivated us to require IND-CCA security, but does IND-CCA always guard against such attacks?
- The decryption algorithm can have **multiple checks** that may cause it to fail. Knowledge of which check failed may convey more information to the adversary.
- Distinguishable decryption failures enabled attacks against TLS **[CHVV 03]**, DTLS **[AP 12]**, and IPsec **[DP 10]**.

Attacks Based on Decryption Failures

- The classic examples are Bleichenbacher's attack on RSA and Vaudenay's padding oracle attack on CBC encryption.

- These attacks motivated us to require IND-CCA security, but does IND-CCA always guard against such attacks?

- The decryption algorithm can have **multiple checks** that may cause it to fail. Knowledge of which check failed may convey more information to the adversary.

- Distinguishable decryption failures enabled attacks against TLS **[CHVV 03]**, DTLS **[AP 12]**, and IPsec **[DP 10]**.

- **GAP:** In IND-CCA the adversary only learns whether a ciphetext is valid or not (distinct decryption failures always return $\perp$).

A Common Response

- *"This is a flaw in the implementation. It can be easily fixed by ensuring that errors are not distinguishable."*
- But errors are useful for troubleshooting; moreover side-channels due to timing or interaction with other protocols (e.g. IPsec) are hard to prevent.

A Common Response

- *"This is a flaw in the implementation. It can be easily fixed by ensuring that errors are not distinguishable."*

- But errors are useful for troubleshooting; moreover side-channels due to timing or interaction with other protocols (e.g. IPsec) are hard to prevent.

- On the other hand it is easy to model distinguishable decryption failures – **multiple-error schemes**.

$$\mathcal{D} : \mathcal{K} \times \mathcal{C} \to \mathcal{M} \cup \mathcal{S}_\perp$$

where $\mathcal{S}_\perp = \{\perp_1, \perp_2, \ldots, \perp_n\}$

- How does this affect the theory of symmetric encryption?

Revisiting Classic Relations

- The following relation is attributed to Bellare and Namprempre **[BN00]**, and to Katz and Yung **[KY00]**.

$$\text{IND-CPA} \wedge \text{INT-CTXT} \Rightarrow \text{IND-CCA}$$

Revisiting Classic Relations

- The following relation is attributed to Bellare and Namprempre **[BN00]**, and to Katz and Yung **[KY00]**.

$$\text{IND-CPA} \wedge \text{INT-CTXT} \Rightarrow \text{IND-CCA}$$

- This relation provides a simple technique for realizing IND-CCA secure schemes in the symmetric setting.

- Furthermore INT-CTXT $+$ IND-CPA has become the target security notion for **authenticated encryption**, since INT-CTXT $\Rightarrow$ INT-PTXT.

Revisiting Classic Relations

- In their work on SSH, Bellare, Kohno, and Namprempre **[BKN04]** extended this relation to the stateful setting.

$$\text{IND-CPA} \wedge \text{INT-sfCTXT} \Rightarrow \text{IND-sfCCA}$$

- INT-sfCTXT and IND-sfCCA are strengthened variations, which additionally capture replay and reordering attacks.

- Any encryption scheme which satisfies these notions must be stateful – hence the name.

Classic Relations in the Multiple-Error Setting

Classic Relations in the Multiple-Error Setting

Theorem

If pseudorandom functions exist, then there exists a multiple-error encryption scheme that is both IND-CPA and INT-CTXT secure, but not IND-CCA secure.

$$\textit{IND-CPA} \wedge \textit{INT-CTXT} \not\Rightarrow \textit{IND-CCA}$$

Classic Relations in the Multiple-Error Setting

Theorem

If pseudorandom functions exist, then there exists a multiple-error encryption scheme that is both IND-CPA and INT-CTXT secure, but not IND-CCA secure.

$$\mathit{IND\text{-}CPA} \wedge \mathit{INT\text{-}CTXT} \not\Rightarrow \mathit{IND\text{-}CCA}$$

- A similar separation holds for the **stateful setting**:

$$\text{IND-CPA} \wedge \text{INT-sfCTXT} \not\Rightarrow \text{IND-sfCCA}$$

- As we shall see, **it is possible to define ciphertext integrity in two ways**, both separations allow the stronger variant.

New Relations in the Multiple-Error Setting

- Given the utility of these relations, an obvious question is whether we can obtain something similar in the multiple-error setting.

New Relations in the Multiple-Error Setting

- Given the utility of these relations, an obvious question is whether we can obtain something similar in the multiple-error setting.

$$\text{IND-CVA} \wedge \text{INT-CTXT} \Rightarrow \text{IND-CCA}$$

New Relations in the Multiple-Error Setting

- Given the utility of these relations, an obvious question is whether we can obtain something similar in the multiple-error setting.

$$\text{IND-CVA} \wedge \text{INT-CTXT} \Rightarrow \text{IND-CCA}$$

- Informally, IND-CVA is described as the IND-CPA game with additional access to a **ciphertext validity oracle** which returns decryption errors but no plaintext.

- The **stronger variant** of ciphertext integrity is **required**.

- Similar relations can be obtained for IND-sfCCA, IND$-CCA, and IND$-sfCCA.

Defining Ciphertext Integrity

INT-CTXT* (weaker variant):

$\mathbf{Exp}_{\mathcal{SE}}^{\text{int-ctxt}*}(\mathcal{A})$	$\text{Enc}(m)$	$\text{Try}^*(c)$
$K \leftarrow \mathcal{K}$ $\mathtt{C} \leftarrow \emptyset, \text{win} \leftarrow 0$ $\mathcal{A}^{\text{Enc}(\cdot),\text{Try}^*(\cdot)}$ **return** win	$c \leftarrow \mathcal{E}_K(m)$ $\mathtt{C} \leftarrow \mathtt{C} \cup c$ **return** c	$m \leftarrow \mathcal{D}_K(c)$ **if** $c \notin \mathtt{C}$ **and** $m \in \mathcal{M}$ **then** win $\leftarrow \mathtt{true}$ **if** $m \in \mathcal{M}$ **then** $m \leftarrow \mathtt{valid}$ **else** $m \leftarrow \mathtt{invalid}$ **return** m

- Try queries reveal only whether a ciphertext is **valid** or **not**.

Defining Ciphertext Integrity

INT-CTXT (stronger variant):

$\underline{\mathbf{Exp}^{\text{int-ctxt}}_{\mathcal{SE}}(\mathcal{A})}$	$\underline{\mathrm{Enc}(m)}$	$\underline{\mathrm{Try}(c)}$
$K \leftarrow \mathcal{K}$	$c \leftarrow \mathcal{E}_K(m)$	$m \leftarrow \mathcal{D}_K(c)$
$\mathtt{C} \leftarrow \emptyset, \mathrm{win} \leftarrow 0$	$\mathtt{C} \leftarrow \mathtt{C} \cup c$	**if** $c \notin \mathtt{C}$ **and** $m \in \mathcal{M}$
$\mathcal{A}^{\mathrm{Enc}(\cdot),\mathrm{Try}(\cdot)}$	**return** c	**then** $\mathrm{win} \leftarrow \mathtt{true}$
return win		**if** $m \in \mathcal{M}$ **then** $m \leftarrow \mathtt{valid}$
		return m

- Try queries reveal either that a ciphertext is **valid** or the **error** that it generates.

Ciphertext Integrity

- Obviously INT-CTXT $\Rightarrow$ INT-CTXT*, but is the converse true?

- The new relations required strong ciphertext integrity, is this necessary or is it just an artefact of the proof?

Ciphertext Integrity

- Obviously INT-CTXT $\Rightarrow$ INT-CTXT*, but is the converse true?
- The new relations required strong ciphertext integrity, is this necessary or is it just an artefact of the proof?
- Both questions are settled through the following non-trivial separation.

Ciphertext Integrity

- Obviously INT-CTXT $\Rightarrow$ INT-CTXT*, but is the converse true? NO
- The new relations required strong ciphertext integrity, is this necessary or is it just an artefact of the proof? NECESSARY
- Both questions are settled through the following non-trivial separation.

Theorem

Given a scheme with a sufficiently large message space that is both IND-CVA and INT-CTXT, we can construct a multiple-error scheme that is both IND-CVA and INT-CTXT* but not IND-CCA.*

$$\textit{IND-CVA} \wedge \textit{INT-CTXT*} \not\Rightarrow \textit{IND-CCA}$$

IND-CCA3

- Rogaway and Shrimpton **[RS06]** introduced a notion that captures concisely the goal for authenticated encryption:

$$\text{IND-CCA3} \Leftrightarrow \text{IND-CPA} \wedge \text{INT-CTXT}.$$

IND-CCA3

- Rogaway and Shrimpton **[RS06]** introduced a notion that captures concisely the goal for authenticated encryption:

$$\text{IND-CCA3} \Leftrightarrow \text{IND-CPA} \wedge \text{INT-CTXT}.$$

- For all adversaries $\mathcal{A}$:

$$\Pr\left[\mathcal{A}^{\mathcal{E}_K(\cdot),\mathcal{D}_K(\cdot)} = 1\right] - \Pr\left[\mathcal{A}^{\mathcal{E}_K(\$|\cdot|),\perp(\cdot)} = 1\right] \leq \epsilon.$$

IND-CCA3

- Rogaway and Shrimpton **[RS06]** introduced a notion that captures concisely the goal for authenticated encryption:

$$\text{IND-CCA3} \Leftrightarrow \text{IND-CPA} \wedge \text{INT-CTXT}.$$

- For all adversaries $\mathcal{A}$:

$$\Pr\left[\mathcal{A}^{\mathcal{E}_K(\cdot),\mathcal{D}_K(\cdot)} = 1\right] - \Pr\left[\mathcal{A}^{\mathcal{E}_K(\$|\cdot|),\perp(\cdot)} = 1\right] \leq \epsilon.$$

- **Can we extend** this notion to the multiple-error setting? **What security** would it guarantee?

IND-CCA3 in the Multiple-Error Setting

- There exists a $\perp_0 \in \mathcal{S}_\perp$ such that for all adversaries $\mathcal{A}$:

$$\Pr\left[\mathcal{A}^{\mathcal{E}_K(\cdot),\mathcal{D}_K(\cdot)} = 1\right] - \Pr\left[\mathcal{A}^{\mathcal{E}_K(\$|\cdot|),\perp_0(\cdot)} = 1\right] \leq \epsilon.$$

IND-CCA3 in the Multiple-Error Setting

- There exists a $\perp_0 \in \mathcal{S}_\perp$ such that for all adversaries $\mathcal{A}$:

$$\Pr\left[\mathcal{A}^{\mathcal{E}_K(\cdot),\mathcal{D}_K(\cdot)} = 1\right] - \Pr\left[\mathcal{A}^{\mathcal{E}_K(\$|\cdot|),\perp_0(\cdot)} = 1\right] \leq \epsilon.$$

- IND-CCA3 provides the following security guarantees:

$$\text{IND-CCA3} \Leftrightarrow \text{IND-CPA} \wedge \text{INT-CTXT*} \wedge \text{INV-ERR}.$$

Informally INV-ERR says that all invalid ciphertexts that an adversary can come up with, will generate the **same error**.

IND-CCA3 in the Multiple-Error Setting

- There exists a $\perp_0 \in \mathcal{S}_\perp$ such that for all adversaries $\mathcal{A}$:

$$\Pr\left[\mathcal{A}^{\mathcal{E}_K(\cdot),\mathcal{D}_K(\cdot)} = 1\right] - \Pr\left[\mathcal{A}^{\mathcal{E}_K(\$|\cdot|),\perp_0(\cdot)} = 1\right] \leq \epsilon.$$

- IND-CCA3 provides the following security guarantees:

$$\text{IND-CCA3} \Leftrightarrow \text{IND-CPA} \wedge \text{INT-CTXT*} \wedge \text{INV-ERR}.$$

  Informally INV-ERR says that all invalid ciphertexts that an adversary can come up with, will generate the **same error**.

- It can further be shown that:

$$\text{IND-CCA3} \Rightarrow \text{IND-CVA} \wedge \text{INT-CTXT} \Rightarrow \text{IND-CCA}.$$

  Hence IND-CCA3 still constitutes a **good notion for authenticated encryption**, albeit perhaps it is too strong.

Authenticated Encryption Through Generic Composition

- In **[BN00]** Encrypt-then-MAC emerges as the preferred generic composition for realizing authenticated encryption.

- Krawczyk **[Kra01]** however, showed that MAC-then-Encrypt is also IND-CCA secure when encryption is instantiated with CBC mode or CTR mode.

Authenticated Encryption Through Generic Composition

- In **[BN00]** Encrypt-then-MAC emerges as the preferred generic composition for realizing authenticated encryption.
- Krawczyk **[Kra01]** however, showed that MAC-then-Encrypt is also IND-CCA secure when encryption is instantiated with CBC mode or CTR mode.
- Hence, when encryption is instantiated with CBC mode or CTR mode, the question as to which generic composition is better remains open.
- Nonetheless practical cryptosystems (using CBC and CTR) based on EtM have proved to be less vulnerable to attack than ones based on MtE.

Re-examining Generic Compositions

- Re-examining generic compositions in the light of distinguishable decryption failures, provides new formal evidence to support this observation.

- We consider an Encode-then-Encrypt-then-MAC (EEM) composition – to account for the pre-processing that is common in practical schemes.

Re-examining Generic Compositions

- Re-examining generic compositions in the light of distinguishable decryption failures, provides new formal evidence to support this observation.

- We consider an Encode-then-Encrypt-then-MAC (EEM) composition – to account for the pre-processing that is common in practical schemes.

Theorem

For any multiple-error encoding scheme, any IND-CPA multiple-error encryption scheme, and any UF-CMA MAC, the EEM composition yields an IND-CCA3 secure scheme.

Re-examining Generic Compositions

- This theorem says that EEM is a **robust composition**, since security holds even when decryption failures are distinguishable, and without assuming anything about the error behaviour of the encoding or encryption components.

Re-examining Generic Compositions

- This theorem says that EEM is a **robust composition**, since security holds even when decryption failures are distinguishable, and without assuming anything about the error behaviour of the encoding or encryption components.

- Attacks on SSL/TLS [CHVV03], IPsec [DP10], and DTLS [AP12] serve as counterexamples that similar general statements cannot be made about MAC-then-Encode-then-Encrypt.

Re-examining Generic Compositions

- This theorem says that EEM is a **robust composition**, since security holds even when decryption failures are distinguishable, and without assuming anything about the error behaviour of the encoding or encryption components.
- Attacks on SSL/TLS [CHVV03], IPsec [DP10], and DTLS [AP12] serve as counterexamples that similar general statements cannot be made about MAC-then-Encode-then-Encrypt.
- It may seem unfair that we do not consider multiple-error MACs. This is justified as follows:
  - Most MACs verify the tag by recomputing the tag and comparing – only one test condition.
  - When this is implemented badly (the keyczar library example) it results in the MAC itself not being secure.

Conclusion

- We propose the multiple-error setting in order to obtain security guarantees that are **more relevant to practice**.

Conclusion

- We propose the multiple-error setting in order to obtain security guarantees that are **more relevant to practice**.
- **Preventive Approach:** Assign distinct error messages to the distinct checks made during decryption $\Rightarrow$ achieve security that is **less implementation-dependent**.

Conclusion

- We propose the multiple-error setting in order to obtain security guarantees that are **more relevant to practice**.
- **Preventive Approach:** Assign distinct error messages to the distinct checks made during decryption $\Rightarrow$ achieve security that is **less implementation-dependent**.
- **A Posteriori Analysis:** Alternatively the multiple-error setting can be used to model realizations of cryptographic protocols and analyze the **security of the implementation**.