On the confidentiality and integrity of ECIES scheme

Kirill Tsaregorodtsev
Researcher at Cryptography laboratory,
JSRPC ``Kryptonite”, Moscow, Russia

CTCrypt’2023

КРИПТОНИТ

Outline

1. Introduction
2. The object of study: ECIES scheme
3. Security models
4. Main results

Table of Contents

Where does it come from?

- Analysis of 5G protocols.

Where does it come from?

- Analysis of 5G protocols.
- The very first step of 5G-AKA (auth. key agreement protocol) is to send a unique identifier of the User to the Home Network.

Where does it come from?

- Analysis of 5G protocols.
- The very first step of 5G-AKA (auth. key agreement protocol) is to send a unique identifier of the User to the Home Network.
- We want **user privacy**.

Where does it come from?

- Analysis of 5G protocols.
- The very first step of 5G-AKA (auth. key agreement protocol) is to send a unique identifier of the User to the Home Network.
- We want **user privacy**.
- This property implies at least message confidentiality and integrity of the ECIES scheme in the “multiple queries” setting (but may be more, e.g., different error codes…).

5G-AKA

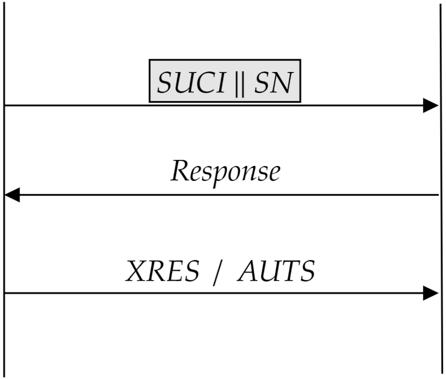
- Key agreement protocol based on a pre-shared secret keys.

5G-AKA

- Key agreement protocol based on a pre-shared secret keys.
- Main part of the protocol: three messages.

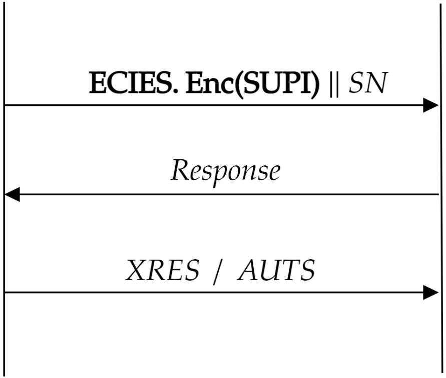
5G-AKA

- Key agreement protocol based on a pre-shared secret keys.
- Main part of the protocol: three messages.



5G-AKA: focus on ECIES

- Key agreement protocol based on a pre-shared secret keys.
- Main part of the protocol: three messages.



ECIES scheme

- Hybrid encryption scheme (key exchange + authenticated encryption).

[1] Gayoso Martínez, Hernández Encinas, and Queiruga Dios, “Security and practical considerations when implementing the elliptic curve integrated encryption scheme”; Martínez, Encinas, et al., “A comparison of the standardized versions of ECIES”; Shoup, *A Proposal for an ISO Standard for Public Key Encryption*.

ECIES scheme

- Hybrid encryption scheme (key exchange + authenticated encryption).
- Should provide confidentiality and integrity of messages (more on that later).

[1] Gayoso Martínez, Hernández Encinas, and Queiruga Dios, "Security and practical considerations when implementing the elliptic curve integrated encryption scheme"; Martínez, Encinas, et al., "A comparison of the standardized versions of ECIES"; Shoup, *A Proposal for an ISO Standard for Public Key Encryption*.

ECIES scheme

- Hybrid encryption scheme (key exchange + authenticated encryption).
- Should provide confidentiality and integrity of messages (more on that later).
- Widely standardized and deployed[1].

[1] Gayoso Martínez, Hernández Encinas, and Queiruga Dios, “Security and practical considerations when implementing the elliptic curve integrated encryption scheme”; Martínez, Encinas, et al., “A comparison of the standardized versions of ECIES”; Shoup, *A Proposal for an ISO Standard for Public Key Encryption*.

ECIES scheme

- Hybrid encryption scheme (key exchange + authenticated encryption).
- Should provide confidentiality and integrity of messages (more on that later).
- Widely standardized and deployed[1].
- In this work we describe it slightly more general than it is standardized based on “abstract” authenticated encryption scheme $\mathcal{AE}$ (AE-scheme) and key exchange scheme $\mathcal{KE}$ (KE-scheme).

[1] Gayoso Martínez, Hernández Encinas, and Queiruga Dios, “Security and practical considerations when implementing the elliptic curve integrated encryption scheme”; Martínez, Encinas, et al., “A comparison of the standardized versions of ECIES”; Shoup, *A Proposal for an ISO Standard for Public Key Encryption*.

Previous analysis

- Confidentiality is analyzed in the $\mathrm{LOR\text{-}CCA}$ model with **only one** encryption challenge query[2].

[2] Abdalla, Bellare, and Rogaway, "The oracle Diffie-Hellman assumptions and an analysis of DHIES"; Shoup, *A Proposal for an ISO Standard for Public Key Encryption*; Smart, "The exact security of ECIES in the generic group model."

[3] Bellare and Namprempre, "Authenticated encryption: Relations among notions and analysis of the generic composition paradigm."

Previous analysis

- Confidentiality is analyzed in the $\mathrm{LOR\text{-}CCA}$ model with **only one** encryption challenge query[2].
- It seems that integrity was not analyzed for some reasons ($\mathrm{INT\text{-}CTXT}$? $\mathrm{INT\text{-}PTXT}$?)[3].

[2] Abdalla, Bellare, and Rogaway, "The oracle Diffie-Hellman assumptions and an analysis of DHIES"; Shoup, *A Proposal for an ISO Standard for Public Key Encryption*; Smart, "The exact security of ECIES in the generic group model."

[3] Bellare and Namprempre, "Authenticated encryption: Relations among notions and analysis of the generic composition paradigm."

Previous analysis

- Confidentiality is analyzed in the $\mathrm{LOR\text{-}CCA}$ model with **only one** encryption challenge query[2].
- It seems that integrity was not analyzed for some reasons ($\mathrm{INT\text{-}CTXT}$? $\mathrm{INT\text{-}PTXT}$?)[3].
- Only for the concrete standardized scheme: Encrypt-then-MAC, key exchange based on Diffie-Hellman-like approach (instead of more general treatment with any AE/KE-scheme).

[2] Abdalla, Bellare, and Rogaway, "The oracle Diffie-Hellman assumptions and an analysis of DHIES"; Shoup, *A Proposal for an ISO Standard for Public Key Encryption*; Smart, "The exact security of ECIES in the generic group model."

[3] Bellare and Namprempre, "Authenticated encryption: Relations among notions and analysis of the generic composition paradigm."

Our goals

- Analyze confidentiality and integrity in the "usual" $\mathrm{LOR\text{-}CCA}$ (conf.) and $\mathrm{INT\text{-}CTXT}$ (integr.) models.

[4] Alekseev et al., "On the cryptographic properties of algorithms accompanying the applications of standards GOST R 34.11-2012 and GOST R 34.10-2012."

Our goals

- Analyze confidentiality and integrity in the "usual" $\mathrm{LOR\text{-}CCA}$ (conf.) and $\mathrm{INT\text{-}CTXT}$ (integr.) models.
- In the general setting ("generic" key exchange scheme (more on that later) and AE(AD)-scheme).

[4] Alekseev et al., "On the cryptographic properties of algorithms accompanying the applications of standards GOST R 34.11-2012 and GOST R 34.10-2012."

Our goals

- Analyze confidentiality and integrity in the “usual” $\mathrm{LOR\text{-}CCA}$ (conf.) and $\mathrm{INT\text{-}CTXT}$ (integr.) models.
- In the general setting (“generic” key exchange scheme (more on that later) and AE(AD)-scheme).
- Draw conclusions for the case when ECIES is instantiated with Russian crypto-algorithms (such as VKO scheme[4]).

[4] Alekseev et al., “On the cryptographic properties of algorithms accompanying the applications of standards GOST R 34.11-2012 and GOST R 34.10-2012.”

Table of Contents

Building blocks

Firstly we have to discuss two main building blocks of the scheme:

- authenticated encryption scheme $\mathcal{AE}$ (AE-scheme);
- key exchange scheme $\mathcal{KE}$ (KE-scheme).

Authenticated encryption scheme

Triplet $\mathcal{AE} = (\mathbf{KeyGen}, \mathbf{Enc}, \mathbf{Dec})$ of (probabilistic) algorithms:

[5] Akhmetzyanova et al., “Security of Multilinear Galois Mode (MGM)”; Nozdrunov, “Parallel and double block cipher mode of operation (PD–mode) for authenticated encryption.”

[6] Bellare and Namprempre, “Authenticated encryption: Relations among notions and analysis of the generic composition paradigm.”

Authenticated encryption scheme

Triplet $\mathcal{AE} = (\mathbf{KeyGen}, \mathbf{Enc}, \mathbf{Dec})$ of (probabilistic) algorithms:

- key generation algorithm **KeyGen**; no input, returns a randomly chosen key k (e.g., from the set $\{0,1\}^{klen}$);

[5] Akhmetzyanova et al., "Security of Multilinear Galois Mode (MGM)"; Nozdrunov, "Parallel and double block cipher mode of operation (PD–mode) for authenticated encryption."

[6] Bellare and Namprempre, "Authenticated encryption: Relations among notions and analysis of the generic composition paradigm."

Authenticated encryption scheme

Triplet $\mathcal{AE} = (\mathbf{KeyGen}, \mathbf{Enc}, \mathbf{Dec})$ of (probabilistic) algorithms:

- key generation algorithm **KeyGen**; no input, returns a randomly chosen key k (e.g., from the set $\{0,1\}^{klen}$);
- encryption algorithm **Enc**; input: key k and the message m, returns a ciphertext $ct \xleftarrow{\$} \mathcal{AE}.\mathbf{Enc}(k, m)$;

[5] Akhmetzyanova et al., "Security of Multilinear Galois Mode (MGM)"; Nozdrunov, "Parallel and double block cipher mode of operation (PD–mode) for authenticated encryption."

[6] Bellare and Namprempre, "Authenticated encryption: Relations among notions and analysis of the generic composition paradigm."

Authenticated encryption scheme

Triplet $\mathcal{AE} = (\mathbf{KeyGen}, \mathbf{Enc}, \mathbf{Dec})$ of (probabilistic) algorithms:

- key generation algorithm **KeyGen**; no input, returns a randomly chosen key k (e.g., from the set $\{0,1\}^{klen}$);
- encryption algorithm **Enc**; input: key k and the message m, returns a ciphertext $ct \xleftarrow{\$} \mathcal{AE}.\mathbf{Enc}(k, m)$;
- decryption algorithm **Dec**; input: key k and the ciphertext ct, returns $m \leftarrow \mathcal{AE}.\mathbf{Dec}(k, ct)$, which is either some message, or the special decryption error symbol $\perp$.

[5] Akhmetzyanova et al., "Security of Multilinear Galois Mode (MGM)"; Nozdrunov, "Parallel and double block cipher mode of operation (PD–mode) for authenticated encryption."
[6] Bellare and Namprempre, "Authenticated encryption: Relations among notions and analysis of the generic composition paradigm."

Authenticated encryption scheme

Triplet $\mathcal{AE} = (\mathbf{KeyGen}, \mathbf{Enc}, \mathbf{Dec})$ of (probabilistic) algorithms:

- key generation algorithm **KeyGen**; no input, returns a randomly chosen key k (e.g., from the set $\{0,1\}^{klen}$);
- encryption algorithm **Enc**; input: key k and the message m, returns a ciphertext $ct \xleftarrow{\$} \mathcal{AE}.\mathbf{Enc}(k, m)$;
- decryption algorithm **Dec**; input: key k and the ciphertext ct, returns $m \leftarrow \mathcal{AE}.\mathbf{Dec}(k, ct)$, which is either some message, or the special decryption error symbol $\perp$.

Correct decryption: for any m, any $k \xleftarrow{\$} \mathcal{AE}.\mathbf{KeyGen}$: $\mathcal{AE}.\mathbf{Dec}(k, \mathcal{AE}.\mathbf{Enc}(k, m)) = m$.

[5] Akhmetzyanova et al., "Security of Multilinear Galois Mode (MGM)"; Nozdrunov, "Parallel and double block cipher mode of operation (PD–mode) for authenticated encryption."

[6] Bellare and Namprempre, "Authenticated encryption: Relations among notions and analysis of the generic composition paradigm."

Authenticated encryption scheme

Triplet $\mathcal{AE} = (\textbf{KeyGen}, \textbf{Enc}, \textbf{Dec})$ of (probabilistic) algorithms:

- key generation algorithm **KeyGen**; no input, returns a randomly chosen key k (e.g., from the set $\{0,1\}^{klen}$);
- encryption algorithm **Enc**; input: key k and the message m, returns a ciphertext $ct \xleftarrow{\$} \mathcal{AE}.\textbf{Enc}(k, m)$;
- decryption algorithm **Dec**; input: key k and the ciphertext ct, returns $m \leftarrow \mathcal{AE}.\textbf{Dec}(k, ct)$, which is either some message, or the special decryption error symbol $\perp$.

Correct decryption: for any m, any $k \xleftarrow{\$} \mathcal{AE}.\textbf{KeyGen}$: $\mathcal{AE}.\textbf{Dec}(k, \mathcal{AE}.\textbf{Enc}(k, m)) = m$.

Examples: MGM mode[5]; CTR + CMAC, EtM[6].

[5] Akhmetzyanova et al., “Security of Multilinear Galois Mode (MGM)”; Nozdrunov, “Parallel and double block cipher mode of operation (PD–mode) for authenticated encryption.”

[6] Bellare and Namprempre, “Authenticated encryption: Relations among notions and analysis of the generic composition paradigm.”

Key exchange scheme

Pair of algorithms $\mathcal{KE} = (\texttt{KeyPairGen}, \texttt{Combine})$:

[7] Alekseev et al., "On the cryptographic properties of algorithms accompanying the applications of standards GOST R 34.11-2012 and GOST R 34.10-2012."

Key exchange scheme

Pair of algorithms $\mathcal{KE} = (\mathtt{KeyPairGen}, \mathtt{Combine})$:

- private-public key pair generation algorithm **`KeyPairGen`**; no input, returns a randomly chosen key pair (sk, pk);

[7] Alekseev et al., "On the cryptographic properties of algorithms accompanying the applications of standards GOST R 34.11-2012 and GOST R 34.10-2012."

Key exchange scheme

Pair of algorithms $\mathcal{KE} = (\texttt{KeyPairGen}, \texttt{Combine})$:

- private-public key pair generation algorithm **`KeyPairGen`**; no input, returns a randomly chosen key pair (sk, pk);
- shared secret value generation algorithm **`Combine`**; input: private key sk, public key pk, returns generated shared secret k.

[7] Alekseev et al., "On the cryptographic properties of algorithms accompanying the applications of standards GOST R 34.11-2012 and GOST R 34.10-2012."

Key exchange scheme

Pair of algorithms $\mathcal{KE} = (\texttt{KeyPairGen}, \texttt{Combine})$:

- private-public key pair generation algorithm $\texttt{KeyPairGen}$; no input, returns a randomly chosen key pair (sk, pk);
- shared secret value generation algorithm $\texttt{Combine}$; input: private key sk, public key pk, returns generated shared secret k.

Correct shared secret generation requirement: for any two key pairs $(sk, pk) \xleftarrow{\$} \mathcal{KE}.\texttt{KeyPairGen}$ and $(esk, epk) \xleftarrow{\$} \mathcal{KE}.\texttt{KeyPairGen}$:

$$\mathcal{KE}.\texttt{Combine}(sk, epk) = \mathcal{KE}.\texttt{Combine}(esk, pk).$$

[7] Alekseev et al., "On the cryptographic properties of algorithms accompanying the applications of standards GOST R 34.11-2012 and GOST R 34.10-2012."

Key exchange scheme

Pair of algorithms $\mathcal{KE} = (\texttt{KeyPairGen}, \texttt{Combine})$:

- private-public key pair generation algorithm $\texttt{KeyPairGen}$; no input, returns a randomly chosen key pair (sk, pk);
- shared secret value generation algorithm $\texttt{Combine}$; input: private key sk, public key pk, returns generated shared secret k.

Correct shared secret generation requirement: for any two key pairs $(sk, pk) \xleftarrow{\$} \mathcal{KE}.\texttt{KeyPairGen}$ and $(esk, epk) \xleftarrow{\$} \mathcal{KE}.\texttt{KeyPairGen}$:

$$\mathcal{KE}.\texttt{Combine}(sk, epk) = \mathcal{KE}.\texttt{Combine}(esk, pk).$$

Example: VKO scheme[7].

[7] Alekseev et al., “On the cryptographic properties of algorithms accompanying the applications of standards GOST R 34.11-2012 and GOST R 34.10-2012.”

ECIES scheme

Alice wants to send message m to Bob, who has a long-term key pair (sk, pk).

ECIES scheme

Alice wants to send message m to Bob, who has a long-term key pair (sk, pk).

Two-step encryption process:

ECIES scheme

Alice wants to send message m to Bob, who has a long-term key pair (sk, pk).

Two-step encryption process:

- generating ephemeral pair $(esk, epk)\mathcal{KE}.\texttt{KeyPairGen}$ and session secret key $k \leftarrow \mathcal{KE}.\texttt{Combine}(esk, pk)$;

Alice wants to send message m to Bob, who has a long-term key pair (sk, pk).

Two-step encryption process:

- generating ephemeral pair $(esk, epk)\mathcal{KE}.\mathtt{KeyPairGen}$ and session secret key $k \leftarrow \mathcal{KE}.\mathtt{Combine}(esk, pk)$;
- encrypting the message m under the key k: $ct \xleftarrow{\$} \mathcal{AE}.\mathtt{Enc}(k, m)$ and sending (epk, ct) to the recipient.

ECIES scheme

Alice wants to send message m to Bob, who has a long-term key pair (sk, pk).

Two-step encryption process:

- generating ephemeral pair $(esk, epk)\mathcal{KE}.\mathtt{KeyPairGen}$ and session secret key $k \leftarrow \mathcal{KE}.\mathtt{Combine}(esk, pk)$;
- encrypting the message m under the key k: $ct \xleftarrow{\$} \mathcal{AE}.\mathtt{Enc}(k, m)$ and sending (epk, ct) to the recipient.

Decryption: generate k using sk and epk, decrypt ct under k.

ECIES scheme

Alice wants to send message m to Bob, who has a long-term key pair (sk, pk).

Two-step encryption process:

- generating ephemeral pair $(esk, epk)\mathcal{KE}.\mathtt{KeyPairGen}$ and session secret key $k \leftarrow \mathcal{KE}.\mathtt{Combine}(esk, pk)$;
- encrypting the message m under the key k: $ct \xleftarrow{\$} \mathcal{AE}.\mathtt{Enc}(k, m)$ and sending (epk, ct) to the recipient.

Decryption: generate k using sk and epk, decrypt ct under k.

Fresh ephemeral key pair on each invocation!

Pseudocode description

$\text{ECIES}.\mathtt{Enc}(pk, m)$	$\text{ECIES}.\mathtt{Dec}(epk, sk, ct)$
$(esk, epk) \xleftarrow{\$} \mathcal{KE}.\mathtt{KeyPairGen}()$	$k \leftarrow \mathcal{KE}.\mathtt{Combine}(sk, epk)$
$k \leftarrow \mathcal{KE}.\mathtt{Combine}(esk, pk)$	$\textbf{return } \mathcal{AE}.\mathtt{Dec}(k, ct)$
$ct \xleftarrow{\$} \mathcal{AE}.\mathtt{Enc}(k, m)$	
$\textbf{return } (epk, ct)$	

Table of Contents

Models: overview

- $\mathrm{LOR\text{-}CCA}$: confidentiality model (for $\mathcal{AE}$ scheme, for ECIES scheme).

Models: overview

- $\mathrm{LOR\text{-}CCA}$: confidentiality model (for $\mathcal{AE}$ scheme, for ECIES scheme).
- $\mathrm{INT\text{-}CTXT}$: integrity model (for $\mathcal{AE}$ scheme, for ECIES scheme).

Models: overview

- $\mathrm{LOR\text{-}CCA}$: confidentiality model (for $\mathcal{AE}$ scheme, for ECIES scheme).
- $\mathrm{INT\text{-}CTXT}$: integrity model (for $\mathcal{AE}$ scheme, for ECIES scheme).
- MODH: key indistinguishability (for $\mathcal{KE}$ scheme).

Models: overview

- $\mathrm{LOR\text{-}CCA}$: confidentiality model (for $\mathcal{AE}$ scheme, for ECIES scheme).
- $\mathrm{INT\text{-}CTXT}$: integrity model (for $\mathcal{AE}$ scheme, for ECIES scheme).
- MODH: key indistinguishability (for $\mathcal{KE}$ scheme).
- Main result-1: $\mathrm{LOR\text{-}CCA}$ for ECIES can be reduced to the $\mathrm{LOR\text{-}CCA}$ for $\mathcal{AE}$ and MODH for $\mathcal{KE}$.

Models: overview

- $\mathrm{LOR\text{-}CCA}$: confidentiality model (for $\mathcal{AE}$ scheme, for ECIES scheme).
- $\mathrm{INT\text{-}CTXT}$: integrity model (for $\mathcal{AE}$ scheme, for ECIES scheme).
- MODH: key indistinguishability (for $\mathcal{KE}$ scheme).
- Main result-1: $\mathrm{LOR\text{-}CCA}$ for ECIES can be reduced to the $\mathrm{LOR\text{-}CCA}$ for $\mathcal{AE}$ and MODH for $\mathcal{KE}$.
- Main result-2: $\mathrm{INT\text{-}CTXT}$ for ECIES can be reduced to the $\mathrm{INT\text{-}CTXT}$ for $\mathcal{AE}$ and MODH for $\mathcal{KE}$.

Confidentiality of $\mathcal{AE}$: $\mathrm{LOR\text{-}CCA}$ model

$\mathrm{LOR\text{-}CCA}$ model (Left-or-Right, Chosen Ciphertext Attack) for the AE-scheme $\mathcal{AE}$ in the multi-user ($D \in \mathbb{N}$) setting.

Confidentiality of $\mathcal{AE}$: $\mathrm{LOR\text{-}CCA}$ model

$\mathrm{LOR\text{-}CCA}$ model (Left-or-Right, Chosen Ciphertext Attack) for the AE-scheme $\mathcal{AE}$ in the multi-user ($D \in \mathbb{N}$) setting.

Interface: two oracles $\mathcal{O}^b_{\mathrm{enc}}$ and $\mathcal{O}_{\mathrm{dec}}$.

Confidentiality of $\mathcal{AE}$: $\mathrm{LOR\text{-}CCA}$ model

$\mathrm{LOR\text{-}CCA}$ model (Left-or-Right, Chosen Ciphertext Attack) for the AE-scheme $\mathcal{AE}$ in the multi-user ($D \in \mathbb{N}$) setting.

Interface: two oracles $\mathcal{O}^b_{\mathrm{enc}}$ and $\mathcal{O}_{\mathrm{dec}}$.

- $\mathcal{O}^b_{\mathrm{enc}}$: takes a triple (i, m_0, m_1) — key index $1 \le i \le D$, message pair (m_0, m_1); returns $ct \xleftarrow{\$} \mathcal{AE}.\mathsf{Enc}(k_i, m_b)$.

Confidentiality of $\mathcal{AE}$: $\mathrm{LOR\text{-}CCA}$ model

$\mathrm{LOR\text{-}CCA}$ model (Left-or-Right, Chosen Ciphertext Attack) for the AE-scheme $\mathcal{AE}$ in the multi-user ($D \in \mathbb{N}$) setting.

Interface: two oracles $\mathcal{O}^b_{\mathrm{enc}}$ and $\mathcal{O}_{\mathrm{dec}}$.

- $\mathcal{O}^b_{\mathrm{enc}}$: takes a triple (i, m_0, m_1) — key index $1 \le i \le D$, message pair (m_0, m_1); returns $ct \xleftarrow{\$} \mathcal{AE}.\mathsf{Enc}(k_i, m_b)$.
- $\mathcal{O}_{\mathrm{dec}}$ takes a pair — key index $1 \le i \le D$, ciphertext ct; if ct was not returned as an answer to the $\mathcal{O}_{\mathrm{enc}}$ query of the type $(i, \cdot, \cdot)$ before, returns $\mathcal{AE}.\mathsf{Dec}(k_i, ct)$, otherwise an error.

Confidentiality of $\mathcal{AE}$: $\mathrm{LOR\text{-}CCA}$ model

$\mathrm{LOR\text{-}CCA}$ model (Left-or-Right, Chosen Ciphertext Attack) for the AE-scheme $\mathcal{AE}$ in the multi-user ($D \in \mathbb{N}$) setting.

Interface: two oracles $\mathcal{O}^b_{\mathrm{enc}}$ and $\mathcal{O}_{\mathrm{dec}}$.

- $\mathcal{O}^b_{\mathrm{enc}}$: takes a triple (i, m_0, m_1) — key index $1 \le i \le D$, message pair (m_0, m_1); returns $ct \xleftarrow{\$} \mathcal{AE}.\mathbf{Enc}(k_i, m_b)$.
- $\mathcal{O}_{\mathrm{dec}}$ takes a pair — key index $1 \le i \le D$, ciphertext ct; if ct was not returned as an answer to the $\mathcal{O}_{\mathrm{enc}}$ query of the type $(i, \cdot, \cdot)$ before, returns $\mathcal{AE}.\mathbf{Dec}(k_i, ct)$, otherwise an error.

Goal: predict the bit b fixed in the $\mathcal{O}^b_{\mathrm{enc}}$.

Confidentiality of $\mathcal{AE}$: $\mathrm{LOR\text{-}CCA}$ model

$\mathrm{LOR\text{-}CCA}$ model (Left-or-Right, Chosen Ciphertext Attack) for the AE-scheme $\mathcal{AE}$ in the multi-user ($D \in \mathbb{N}$) setting.

Interface: two oracles $\mathcal{O}^b_{\mathrm{enc}}$ and $\mathcal{O}_{\mathrm{dec}}$.

- $\mathcal{O}^b_{\mathrm{enc}}$: takes a triple (i, m_0, m_1) — key index $1 \le i \le D$, message pair (m_0, m_1); returns $ct \xleftarrow{\$} \mathcal{AE}.\mathbf{Enc}(k_i, m_b)$.
- $\mathcal{O}_{\mathrm{dec}}$ takes a pair — key index $1 \le i \le D$, ciphertext ct; if ct was not returned as an answer to the $\mathcal{O}_{\mathrm{enc}}$ query of the type $(i, \cdot, \cdot)$ before, returns $\mathcal{AE}.\mathbf{Dec}(k_i, ct)$, otherwise an error.

Goal: predict the bit b fixed in the $\mathcal{O}^b_{\mathrm{enc}}$.

Success measure: advantage

$$\mathbf{Adv}^{\mathrm{LOR\text{-}CCA}}_{\mathcal{AE}}(\mathcal{A}) = \mathbb{P}\left[\mathbf{Exp}^{\mathrm{LOR\text{-}CCA\text{-}1}}_{\mathcal{AE}}(\mathcal{A}) \to 1\right] - \mathbb{P}\left[\mathbf{Exp}^{\mathrm{LOR\text{-}CCA\text{-}0}}_{\mathcal{AE}}(\mathcal{A}) \to 1\right].$$

Confidentiality of $\mathcal{AE}$: pseudocode

$\underline{\mathbf{Exp}^{\text{LOR-CCA-}b}_{\mathcal{AE}}(\mathcal{A})}$

for $1 \leq i \leq D$ **do**

$\quad k_i \xleftarrow{\$} \mathcal{AE}.\mathsf{KeyGen}()$

endfor

$\mathit{sent} \leftarrow [\,]$

$b' \xleftarrow{\$} \mathcal{A}^{\mathcal{O}^b_{\text{enc}}, \mathcal{O}_{\text{dec}}}$

return b'

Confidentiality of $\mathcal{AE}$: pseudocode

$\underline{\mathbf{Exp}^{\text{LOR-CCA-}b}_{\mathcal{AE}}(\mathcal{A})}$

for $1 \leq i \leq D$ **do**

  $k_i \xleftarrow{\$} \mathcal{AE}.\mathsf{KeyGen}()$

endfor

$sent \leftarrow [\,]$

$b' \xleftarrow{\$} \mathcal{A}^{\mathcal{O}^b_{\text{enc}}, \mathcal{O}_{\text{dec}}}$

return b'

$\underline{\mathcal{O}^b_{\text{enc}}(i, m_0, m_1)}$

$ct \xleftarrow{\$} \mathcal{AE}.\mathsf{Enc}(k_i, m_b)$

$sent[i] \leftarrow sent[i] \cup \{ct\}$

return ct

Confidentiality of $\mathcal{AE}$: pseudocode

$$\mathbf{Exp}^{\text{LOR-CCA-}b}_{\mathcal{AE}}(\mathcal{A})$$

$\textbf{for } 1 \le i \le D \textbf{ do}$

$\quad k_i \xleftarrow{\$} \mathcal{AE}.\mathsf{KeyGen}()$

$\textbf{endfor}$

$sent \leftarrow [\,]$

$b' \xleftarrow{\$} \mathcal{A}^{\mathcal{O}^b_{\text{enc}}, \mathcal{O}_{\text{dec}}}$

$\textbf{return } b'$

$$\mathcal{O}^b_{\text{enc}}(i, m_0, m_1)$$

$ct \xleftarrow{\$} \mathcal{AE}.\mathsf{Enc}(k_i, m_b)$

$sent[i] \leftarrow sent[i] \cup \{ct\}$

$\textbf{return } ct$

$$\mathcal{O}_{\text{dec}}(i, ct)$$

$\textbf{if } (ct \in sent[i])$

$\quad \textbf{return } \perp$

$\textbf{fi}$

$\textbf{return } \mathcal{AE}.\mathsf{Dec}(k_i, ct)$

Confidentiality of $\mathcal{AE}$: success measure

$$\mathbf{Adv}_{\mathcal{AE}}^{\text{LOR-CCA}}(t, Q_e, Q_d, L_e, L_d, M_e, M_d; D)$$

the maximal advantage $\mathbf{Adv}_{\mathcal{AE}}^{\text{LOR-CCA}}(\mathcal{A})$; the maximum is over the adversaries $\mathcal{A}$ whose time complexity is at most t and with the following restrictions on oracle queries $(1 \leq i \leq D)$:

Confidentiality of $\mathcal{AE}$: success measure

$$\mathbf{Adv}^{\text{LOR-CCA}}_{\mathcal{AE}}(t, Q_e, Q_d, L_e, L_d, M_e, M_d; D)$$

the maximal advantage $\mathbf{Adv}^{\text{LOR-CCA}}_{\mathcal{AE}}(\mathcal{A})$; the maximum is over the adversaries $\mathcal{A}$ whose time complexity is at most t and with the following restrictions on oracle queries ($1 \leq i \leq D$):

- the number of queries of the type (i, m_0, m_1) to the $\mathcal{O}^b_{\text{enc}}$ oracle ((i, ct) to the $\mathcal{O}_{\text{dec}}$ oracle) does not exceed $Q_e[i]$ ($Q_d[i]$ resp.);
- the total length of the queries $\sum |m_0| = \sum |m_1|$ among queries of the type (i, m_0, m_1) to the $\mathcal{O}^b_{\text{enc}}$ oracle ($\sum |ct|$ among queries of the type (i, ct) to the $\mathcal{O}_{\text{dec}}$ oracle) does not exceed $L_e[i]$ ($L_d[i]$ resp.);
- the maximal length of the query $\max |m_0| = \max |m_1|$ among queries of the type (i, m_0, m_1) to the $\mathcal{O}^b_{\text{enc}}$ oracle ($\max |ct|$ among queries of the type (i, ct) to the $\mathcal{O}_{\text{dec}}$ oracle) does not exceed $M_e[i]$ ($M_d[i]$ resp.).

Confidentiality of ECIES: $\mathrm{LOR\text{-}CCA}$ model

- Essentially the same as for the case of $\mathrm{LOR\text{-}CCA}$ model for $\mathcal{AE}$ scheme.

Confidentiality of ECIES: $\mathrm{LOR\text{-}CCA}$ model

- Essentially the same as for the case of $\mathrm{LOR\text{-}CCA}$ model for $\mathcal{AE}$ scheme.
- Noticeable exceptions: generation of a fresh key during each invokation; "number of parties" D is essentially the same as the total number of queries.

Confidentiality of ECIES: $\mathrm{LOR\text{-}CCA}$ model

- Essentially the same as for the case of $\mathrm{LOR\text{-}CCA}$ model for $\mathcal{AE}$ scheme.
- Noticeable exceptions: generation of a fresh key during each invokation; “number of parties” D is essentially the same as the total number of queries.
- Guarantees: cannot guess with probability “greater” than $\frac{1}{2}$ which plaintext was encrypted.

Confidentiality of ECIES: pseudocode

$\mathbf{Exp}_{\mathrm{ECIES}}^{\mathrm{LOR\text{-}CCA\text{-}}b}(\mathcal{A})$

$(sk, pk) \xleftarrow{\$} \mathcal{KE}.\mathtt{KeyPairGen}()$
$sent \leftarrow [\,]$
$b' \xleftarrow{\$} \mathcal{A}^{\mathcal{O}_{\mathrm{enc}}^{b}, \mathcal{O}_{\mathrm{dec}}}(pk)$
return b'

$\mathcal{O}_{\mathrm{enc}}^{b}(m_0, m_1)$

$(epk, esk) \xleftarrow{\$} \mathcal{KE}.\mathtt{KeyPairGen}()$
$k \leftarrow \mathcal{KE}.\mathtt{Combine}(sk, epk)$
$ct \xleftarrow{\$} \mathcal{AE}.\mathtt{Enc}(k, m_b)$
$sent \leftarrow sent \cup \{(epk, ct)\}$
return (epk, ct)

$\mathcal{O}_{\mathrm{dec}}(epk, ct)$

if $(epk, ct) \in sent$
  return $\perp$
fi
$k \leftarrow \mathcal{KE}.\mathtt{Combine}(sk, epk)$
return $\mathcal{AE}.\mathtt{Dec}(k, ct)$

Confidentiality of ECIES: success measure

$$\mathbf{Adv}_{\mathrm{ECIES}}^{\mathrm{LOR\text{-}CCA}}(t, q_e, q_d, l_e, l_d, \mu_e, \mu_d)$$

maximal advantage $\mathbf{Adv}_{\mathrm{ECIES}}^{\mathrm{LOR\text{-}CCA}}(\mathcal{A})$, where the maximum is taken over the adversaries $\mathcal{A}$ whose time complexity is at most t and with the following restrictions on the oracle queries:

Confidentiality of ECIES: success measure

$$\mathbf{Adv}^{\text{LOR-CCA}}_{\text{ECIES}}(t, q_e, q_d, l_e, l_d, \mu_e, \mu_d)$$

maximal advantage $\mathbf{Adv}^{\text{LOR-CCA}}_{\text{ECIES}}(\mathcal{A})$, where the maximum is taken over the adversaries $\mathcal{A}$ whose time complexity is at most t and with the following restrictions on the oracle queries:

- the number of queries to the $\mathcal{O}^b_{\text{enc}}$ oracle (to the $\mathcal{O}_{\text{dec}}$ oracle) does not exceed q_e (q_d resp.);

Confidentiality of ECIES: success measure

$$\mathbf{Adv}^{\mathrm{LOR\text{-}CCA}}_{\mathrm{ECIES}}(t, q_e, q_d, l_e, l_d, \mu_e, \mu_d)$$

maximal advantage $\mathbf{Adv}^{\mathrm{LOR\text{-}CCA}}_{\mathrm{ECIES}}(\mathcal{A})$, where the maximum is taken over the adversaries $\mathcal{A}$ whose time complexity is at most t and with the following restrictions on the oracle queries:

- the number of queries to the $\mathcal{O}^b_{\mathrm{enc}}$ oracle (to the $\mathcal{O}_{\mathrm{dec}}$ oracle) does not exceed q_e (q_d resp.);
- the total length of the queries $\sum |m_0| = \sum |m_1|$ to the $\mathcal{O}^b_{\mathrm{enc}}$ oracle ($\sum |ct|$ to the $\mathcal{O}_{\mathrm{dec}}$ oracle) does not exceed l_e (l_d resp.);

Confidentiality of ECIES: success measure

$$\mathbf{Adv}^{\text{LOR-CCA}}_{\text{ECIES}}(t, q_e, q_d, l_e, l_d, \mu_e, \mu_d)$$

maximal advantage $\mathbf{Adv}^{\text{LOR-CCA}}_{\text{ECIES}}(\mathcal{A})$, where the maximum is taken over the adversaries $\mathcal{A}$ whose time complexity is at most t and with the following restrictions on the oracle queries:

- the number of queries to the $\mathcal{O}^b_{\text{enc}}$ oracle (to the $\mathcal{O}_{\text{dec}}$ oracle) does not exceed q_e (q_d resp.);
- the total length of the queries $\sum |m_0| = \sum |m_1|$ to the $\mathcal{O}^b_{\text{enc}}$ oracle ($\sum |ct|$ to the $\mathcal{O}_{\text{dec}}$ oracle) does not exceed l_e (l_d resp.);
- the maximal length of the query $\max |m_0| = \max |m_1|$ among queries to the $\mathcal{O}^b_{\text{enc}}$ oracle ($\max |ct|$ among queries to the $\mathcal{O}_{\text{dec}}$ oracle) does not exceed μ_e (μ_d resp.);

Integrity for $\mathcal{AE}$: $\mathrm{INT\text{-}CTXT}$ model

$\mathrm{INT\text{-}CTXT}$ model (Integrity of Ciphertexts) for the AE-scheme $\mathcal{AE}$ in the multi-user ($D \in \mathbb{N}$) setting.

Integrity for $\mathcal{AE}$: $\mathrm{INT\text{-}CTXT}$ model

$\mathrm{INT\text{-}CTXT}$ model (Integrity of Ciphertexts) for the AE-scheme $\mathcal{AE}$ in the multi-user ($D \in \mathbb{N}$) setting.

Interface: two oracles $\mathcal{O}_{\mathrm{enc}}$ and $\mathcal{O}_{\mathrm{verify}}$:

Integrity for $\mathcal{AE}$: $\mathrm{INT\text{-}CTXT}$ model

$\mathrm{INT\text{-}CTXT}$ model (Integrity of Ciphertexts) for the AE-scheme $\mathcal{AE}$ in the multi-user ($D \in \mathbb{N}$) setting.

Interface: two oracles $\mathcal{O}_{\mathrm{enc}}$ and $\mathcal{O}_{\mathrm{verify}}$:

- $\mathcal{O}_{\mathrm{enc}}$: input – key index $1 \le i \le D$, message m; returns $ct \xleftarrow{\$} \mathcal{AE}.\mathsf{Enc}(k_i, m)$.

Integrity for $\mathcal{AE}$: $\mathrm{INT\text{-}CTXT}$ model

$\mathrm{INT\text{-}CTXT}$ model (Integrity of Ciphertexts) for the AE-scheme $\mathcal{AE}$ in the multi-user ($D \in \mathbb{N}$) setting.

Interface: two oracles $\mathcal{O}_{\mathrm{enc}}$ and $\mathcal{O}_{\mathrm{verify}}$:

- $\mathcal{O}_{\mathrm{enc}}$: input — key index $1 \leq i \leq D$, message m; returns $ct \xleftarrow{\$} \mathcal{AE}.\mathsf{Enc}(k_i, m)$.
- $\mathcal{O}_{\mathrm{verify}}$: input — ciphertext ct, key index $1 \leq i \leq D$; decrypts $m \leftarrow \mathcal{AE}.\mathsf{Dec}(k_i, ct)$, returns m; if ct was not returned as an answer to the $\mathcal{O}_{\mathrm{enc}}$ query of the type $(i, \cdot)$ before and $m \neq \perp$ (correct decryption), then sets $win \leftarrow \mathtt{true}$.

Integrity for $\mathcal{AE}$: $\mathrm{INT\text{-}CTXT}$ model

$\mathrm{INT\text{-}CTXT}$ model (Integrity of Ciphertexts) for the AE-scheme $\mathcal{AE}$ in the multi-user ($D \in \mathbb{N}$) setting.

Interface: two oracles $\mathcal{O}_{\mathrm{enc}}$ and $\mathcal{O}_{\mathrm{verify}}$:

- $\mathcal{O}_{\mathrm{enc}}$: input — key index $1 \le i \le D$, message m; returns $ct \xleftarrow{\$} \mathcal{AE}.\mathtt{Enc}(k_i, m)$.
- $\mathcal{O}_{\mathrm{verify}}$: input — ciphertext ct, key index $1 \le i \le D$; decrypts $m \leftarrow \mathcal{AE}.\mathtt{Dec}(k_i, ct)$, returns m; if ct was not returned as an answer to the $\mathcal{O}_{\mathrm{enc}}$ query of the type $(i, \cdot)$ before and $m \neq \perp$ (correct decryption), then sets $win \leftarrow \mathtt{true}$.

Goal: forge fresh ciphertext ct that is decrypted to the correct plaintext.

Integrity for $\mathcal{AE}$: $\mathrm{INT\text{-}CTXT}$ model

$\mathrm{INT\text{-}CTXT}$ model (Integrity of Ciphertexts) for the AE-scheme $\mathcal{AE}$ in the multi-user ($D \in \mathbb{N}$) setting.

Interface: two oracles $\mathcal{O}_{\mathrm{enc}}$ and $\mathcal{O}_{\mathrm{verify}}$:

- $\mathcal{O}_{\mathrm{enc}}$: input — key index $1 \leq i \leq D$, message m; returns $ct \xleftarrow{\$} \mathcal{AE}.\mathsf{Enc}(k_i, m)$.
- $\mathcal{O}_{\mathrm{verify}}$: input — ciphertext ct, key index $1 \leq i \leq D$; decrypts $m \leftarrow \mathcal{AE}.\mathsf{Dec}(k_i, ct)$, returns m; if ct was not returned as an answer to the $\mathcal{O}_{\mathrm{enc}}$ query of the type $(i, \cdot)$ before and $m \neq \perp$ (correct decryption), then sets $win \leftarrow \mathtt{true}$.

Goal: forge fresh ciphertext ct that is decrypted to the correct plaintext.

Success measure: advantage

$$\mathbf{Adv}_{\mathcal{AE}}^{\mathrm{INT\text{-}CTXT}}(\mathcal{A}) = \mathbb{P}\left[\mathbf{Exp}_{\mathcal{AE}}^{\mathrm{INT\text{-}CTXT}}(\mathcal{A}) \to 1\right].$$

Integrity for $\mathcal{AE}$: pseudocode

$\underline{\mathbf{Exp}_{\mathcal{AE}}^{\text{INT-CTXT}}(\mathcal{A})}$

$\mathbf{for}\ 1 \leq i \leq D\ \mathbf{do}$

$\quad k_i \xleftarrow{\$} \mathcal{AE}.\mathsf{KeyGen}$

$\mathbf{endfor}$

$\mathit{sent} \leftarrow [\,]$

$\mathit{win} \leftarrow 0$

$\mathcal{A}^{\mathcal{O}_{\text{enc}}, \mathcal{O}_{\text{verify}}}$

$\mathbf{return}\ \mathit{win}$

Integrity for $\mathcal{AE}$: pseudocode

$\underline{\mathbf{Exp}^{\text{INT-CTXT}}_{\mathcal{AE}}(\mathcal{A})}$

$\mathbf{for}\ 1 \leq i \leq D\ \mathbf{do}$

$\quad k_i \xleftarrow{\$} \mathcal{AE}.\mathsf{KeyGen}$

$\mathbf{endfor}$

$\mathit{sent} \leftarrow [\,]$

$\mathit{win} \leftarrow 0$

$\mathcal{A}^{\mathcal{O}_{\text{enc}}, \mathcal{O}_{\text{verify}}}$

$\mathbf{return}\ \mathit{win}$

$\underline{\mathcal{O}_{\text{enc}}(i, m)}$

$ct \xleftarrow{\$} \mathcal{AE}.\mathsf{Enc}(k_i, m)$

$\mathit{sent}[i] \leftarrow \mathit{sent}[i] \cup \{ct\}$

$\mathbf{return}\ ct$

Integrity for $\mathcal{AE}$: pseudocode

$\mathbf{Exp}_{\mathcal{AE}}^{\text{INT-CTXT}}(\mathcal{A})$

$\textbf{for } 1 \leq i \leq D \textbf{ do}$
$\quad k_i \xleftarrow{\$} \mathcal{AE}.\mathsf{KeyGen}$
$\textbf{endfor}$
$\mathit{sent} \leftarrow [\,]$
$\mathit{win} \leftarrow 0$
$\mathcal{A}^{\mathcal{O}_{\text{enc}}, \mathcal{O}_{\text{verify}}}$
$\textbf{return } \mathit{win}$

$\mathcal{O}_{\text{enc}}(i, m)$

$\mathit{ct} \xleftarrow{\$} \mathcal{AE}.\mathsf{Enc}(k_i, m)$
$\mathit{sent}[i] \leftarrow \mathit{sent}[i] \cup \{\mathit{ct}\}$
$\textbf{return } \mathit{ct}$

$\mathcal{O}_{\text{verify}}(i, \mathit{ct})$

$m \leftarrow \mathcal{AE}.\mathsf{Dec}(k_i, \mathit{ct})$
$\textbf{if } (\mathit{ct} \notin \mathit{sent}[i]) \,\&\, (m \neq \bot)$
$\quad \mathit{win} \leftarrow 1$
$\textbf{fi}$
$\textbf{return } m$

Integrity for $\mathcal{AE}$: success measure

$$\mathbf{Adv}_{\mathcal{AE}}^{\text{INT-CTXT}}(t, Q_e, Q_v, L_e, L_v, M_e, M_v; D)$$

maximal advantage $\mathbf{Adv}_{\mathcal{AE}}^{\text{INT-CTXT}}(\mathcal{A})$, where the maximum is taken over the adversaries $\mathcal{A}$ whose time complexity is at most t and with the following restrictions on oracle queries ($1 \leq i \leq D$):

Integrity for $\mathcal{AE}$: success measure

$$\mathbf{Adv}^{\text{INT-CTXT}}_{\mathcal{AE}}(t, Q_e, Q_v, L_e, L_v, M_e, M_v; D)$$

maximal advantage $\mathbf{Adv}^{\text{INT-CTXT}}_{\mathcal{AE}}(\mathcal{A})$, where the maximum is taken over the adversaries $\mathcal{A}$ whose time complexity is at most t and with the following restrictions on oracle queries ($1 \le i \le D$):

- the number of queries of the type (i, m) to the $\mathcal{O}_{\text{enc}}$ oracle ((i, ct) to the $\mathcal{O}_{\text{verify}}$ oracle) does not exceed $Q_e[i]$ ($Q_v[i]$ resp.);
- the total length of the queries $\sum |m|$ among queries of the type (i, m) to the $\mathcal{O}_{\text{enc}}$ oracle ($\sum |ct|$ among queries of the type (i, ct) to the $\mathcal{O}_{\text{verify}}$ oracle) does not exceed $L_e[i]$ ($L_v[i]$ resp.);
- the maximal length of the query $\max |m|$ among queries of the type (i, m) to the $\mathcal{O}_{\text{enc}}$ oracle ($\max |ct|$ among queries of the type (i, ct) to the $\mathcal{O}_{\text{verify}}$ oracle) does not exceed $M_e[i]$ ($M_v[i]$ resp.);

Integrity of ECIES: $\mathrm{INT\text{-}CTXT}$ model

- Essentially the same as for the case of $\mathrm{INT\text{-}CTXT}$ model for $\mathcal{AE}$ scheme.

- Essentially the same as for the case of $\mathrm{INT\text{-}CTXT}$ model for $\mathcal{AE}$ scheme.
- Noticeable exceptions: generation of a fresh key during each invokation; “number of parties” D is essentially the same as the total number of queries.

Integrity of ECIES: $\mathrm{INT\text{-}CTXT}$ model

- Essentially the same as for the case of $\mathrm{INT\text{-}CTXT}$ model for $\mathcal{AE}$ scheme.
- Noticeable exceptions: generation of a fresh key during each invokation; "number of parties" D is essentially the same as the total number of queries.
- Guarantees: cannot forge a correct ciphertext **given an ephemeral public key** (i.e., the key is chosen by the honest party, the goal is to forge for this particular public key).

Integrity of ECIES: pseudocode

$$\underline{\mathbf{Exp}^{\text{INT-CTXT}}_{\text{ECIES}}(\mathcal{A})}$$

$(sk, pk) \xleftarrow{\$} \mathcal{KE}.\mathtt{KeyPairGen}()$

$sent \leftarrow [\,]$

$win \leftarrow 0$

$\mathcal{A}^{\mathcal{O}_{\text{enc}}, \mathcal{O}_{\text{verify}}}(pk)$

$\textbf{return}\ win$

Integrity of ECIES: pseudocode

$\underline{\mathbf{Exp}^{\text{INT-CTXT}}_{\text{ECIES}}(\mathcal{A})}$

$(sk, pk) \xleftarrow{\$} \mathcal{KE}.\texttt{KeyPairGen}()$

$sent \leftarrow [\,]$

$win \leftarrow 0$

$\mathcal{A}^{\mathcal{O}_{\text{enc}}, \mathcal{O}_{\text{verify}}}(pk)$

$\textbf{return } win$

$\underline{\mathcal{O}_{\text{enc}}(m)}$

$(epk, esk) \xleftarrow{\$} \mathcal{KE}.\texttt{KeyPairGen}()$

$k \leftarrow \mathcal{KE}.\texttt{Combine}(sk, epk)$

$ct \xleftarrow{\$} \mathcal{AE}.\texttt{Enc}(k, m)$

$sent[epk] \leftarrow sent[epk] \cup \{ct\}$

$\textbf{return } (epk, ct)$

Integrity of ECIES: pseudocode

$$\mathbf{Exp}^{\text{INT-CTXT}}_{\text{ECIES}}(\mathcal{A})$$

$(sk, pk) \xleftarrow{\$} \mathcal{KE}.\mathtt{KeyPairGen}()$
$sent \leftarrow [\,]$
$win \leftarrow 0$
$\mathcal{A}^{\mathcal{O}_{\text{enc}}, \mathcal{O}_{\text{verify}}}(pk)$
$\mathbf{return}\ win$

$$\mathcal{O}_{\text{enc}}(m)$$

$(epk, esk) \xleftarrow{\$} \mathcal{KE}.\mathtt{KeyPairGen}()$
$k \leftarrow \mathcal{KE}.\mathtt{Combine}(sk, epk)$
$ct \xleftarrow{\$} \mathcal{AE}.\mathtt{Enc}(k, m)$
$sent[epk] \leftarrow sent[epk] \cup \{ct\}$
$\mathbf{return}\ (epk, ct)$

$$\mathcal{O}_{\text{verify}}(epk, ct)$$

$k \leftarrow \mathcal{KE}.\mathtt{Combine}(sk, epk)$
$m \leftarrow \mathcal{AE}.\mathtt{Dec}(k, ct)$
$t_1 \leftarrow (m \neq \perp)$
$t_2 \leftarrow (sent[epk] \neq \perp)$
$t_3 \leftarrow (ct \notin sent[epk])$
$\mathbf{if}\ t_1 \& t_2 \& t_3$
$\quad win \leftarrow 1$
$\mathbf{fi}$
$\mathbf{return}\ m$

$$\mathbf{Adv}^{\text{INT-CTXT}}_{\text{ECIES}}(t, q_e, q_v, l_e, l_v, \mu_e, \mu_v)$$

maximal advantage $\mathbf{Adv}^{\text{INT-CTXT}}_{\text{ECIES}}(\mathcal{A})$, where the maximum is taken over the adversaries $\mathcal{A}$ whose time complexity is at most t and with the following restrictions on the oracle queries:

Integrity of ECIES: success measure

$$\mathbf{Adv}_{\mathrm{ECIES}}^{\mathrm{INT\text{-}CTXT}}(t, q_e, q_v, l_e, l_v, \mu_e, \mu_v)$$

maximal advantage $\mathbf{Adv}_{\mathrm{ECIES}}^{\mathrm{INT\text{-}CTXT}}(\mathcal{A})$, where the maximum is taken over the adversaries $\mathcal{A}$ whose time complexity is at most t and with the following restrictions on the oracle queries:

- the number of queries to the $\mathcal{O}_{\mathrm{enc}}$ oracle (to the $\mathcal{O}_{\mathrm{verify}}$ oracle) does not exceed q_e (q_v resp.);
- the total length of the queries $\sum |m_0| = \sum |m_1|$ to the $\mathcal{O}_{\mathrm{enc}}$ oracle ($\sum |ct|$ to the $\mathcal{O}_{\mathrm{verify}}$ oracle) does not exceed l_e (l_v resp.);
- the maximal length of the query $\max |m_0| = \max |m_1|$ among queries to the $\mathcal{O}_{\mathrm{enc}}$ oracle ($\max |ct|$ among queries to the $\mathcal{O}_{\mathrm{verify}}$ oracle) does not exceed μ_e (μ_v resp.);

Key secrecy for $\mathcal{KE}$: MODH model

MODH model (multiple oracle Diffie-Hellman[8]) for the key exchange scheme $\mathcal{KE}$.

[8] Abdalla, Bellare, and Rogaway, "The oracle Diffie-Hellman assumptions and an analysis of DHIES."

Key secrecy for $\mathcal{KE}$: MODH model

MODH model (multiple oracle Diffie-Hellman[8]) for the key exchange scheme $\mathcal{KE}$.

Interface: two oracles $\mathcal{O}_{\mathrm{comb}}$ and $\mathcal{O}^{b}_{\mathrm{kgen}}$:

[8] Abdalla, Bellare, and Rogaway, “The oracle Diffie-Hellman assumptions and an analysis of DHIES.”

Key secrecy for $\mathcal{KE}$: MODH model

MODH model (multiple oracle Diffie-Hellman[8]) for the key exchange scheme $\mathcal{KE}$.

Interface: two oracles $\mathcal{O}_{\mathrm{comb}}$ and $\mathcal{O}^{b}_{\mathrm{kgen}}$:

- oracle $\mathcal{O}_{\mathrm{comb}}(epk)$ generates a key via $\mathcal{KE}.\mathtt{Combine}$ function using the ephemeral key epk and long-term key sk.

[8] Abdalla, Bellare, and Rogaway, "The oracle Diffie-Hellman assumptions and an analysis of DHIES."

Key secrecy for $\mathcal{KE}$: MODH model

MODH model (multiple oracle Diffie-Hellman[8]) for the key exchange scheme $\mathcal{KE}$.

Interface: two oracles $\mathcal{O}_{\mathrm{comb}}$ and $\mathcal{O}^{b}_{\mathrm{kgen}}$:

- oracle $\mathcal{O}_{\mathrm{comb}}(epk)$ generates a key via $\mathcal{KE}.$**`Combine`** function using the ephemeral key epk and long-term key sk.
- oracle $\mathcal{O}^{b}_{\mathrm{kgen}}$ generates either random keys of a given length (in case of $b = 0$) or keys generated via key exchange scheme (in case of $b = 1$) with some restrictions that exclude trivial attacks, see below;

[8] Abdalla, Bellare, and Rogaway, "The oracle Diffie-Hellman assumptions and an analysis of DHIES."

Key secrecy for $\mathcal{KE}$: MODH model

MODH model (multiple oracle Diffie-Hellman[8]) for the key exchange scheme $\mathcal{KE}$.

Interface: two oracles $\mathcal{O}_{\mathrm{comb}}$ and $\mathcal{O}^{b}_{\mathrm{kgen}}$:

- oracle $\mathcal{O}_{\mathrm{comb}}(epk)$ generates a key via $\mathcal{KE}$.**`Combine`** function using the ephemeral key epk and long-term key sk.
- oracle $\mathcal{O}^{b}_{\mathrm{kgen}}$ generates either random keys of a given length (in case of $b = 0$) or keys generated via key exchange scheme (in case of $b = 1$) with some restrictions that exclude trivial attacks, see below;

Goal: guess the bit b.

[8] Abdalla, Bellare, and Rogaway, "The oracle Diffie-Hellman assumptions and an analysis of DHIES."

Key secrecy for $\mathcal{KE}$: MODH model

MODH model (multiple oracle Diffie-Hellman[8]) for the key exchange scheme $\mathcal{KE}$.

Interface: two oracles $\mathcal{O}_{\mathrm{comb}}$ and $\mathcal{O}^{b}_{\mathrm{kgen}}$:

- oracle $\mathcal{O}_{\mathrm{comb}}(epk)$ generates a key via $\mathcal{KE}$.**`Combine`** function using the ephemeral key epk and long-term key sk.
- oracle $\mathcal{O}^{b}_{\mathrm{kgen}}$ generates either random keys of a given length (in case of $b = 0$) or keys generated via key exchange scheme (in case of $b = 1$) with some restrictions that exclude trivial attacks, see below;

Goal: guess the bit b.

Success measure:

$$\mathbf{Adv}^{\mathrm{MODH}}_{\mathcal{KE}}(\mathcal{A}) = \mathbb{P}\Big[\mathbf{Exp}^{\mathrm{MODH\text{-}1}}_{\mathcal{KE}}(\mathcal{A}) \to 1\Big] - \mathbb{P}\Big[\mathbf{Exp}^{\mathrm{MODH\text{-}0}}_{\mathcal{KE}}(\mathcal{A}) \to 1\Big].$$

[8] Abdalla, Bellare, and Rogaway, "The oracle Diffie-Hellman assumptions and an analysis of DHIES."

Key secrecy for $\mathcal{KE}$: pseudocode

$\mathbf{Exp}_{\mathcal{KE}}^{\text{MODH-}b}(\mathcal{A})$

$(sk, pk) \xleftarrow{\$} \mathcal{KE}.\mathtt{KeyPairGen}()$

$Keys \leftarrow [\,]$

$b' \xleftarrow{\$} \mathcal{A}^{\mathcal{O}^b_{\text{kgen}}, \mathcal{O}_{\text{comb}}}(pk)$

$\mathbf{return}\ b'$

$\mathcal{O}_{\text{comb}}(epk)$

$\mathbf{if}\ Keys[epk] = \bot$

$\quad \mathbf{return}\ \mathcal{KE}.\mathtt{Combine}(sk, epk)$

$\mathbf{else}$

$\quad \mathbf{return}\ Keys[epk]$

$\mathbf{fi}$

$\mathcal{O}^b_{\text{kgen}}()$

$(esk, epk) \xleftarrow{\$} \mathcal{KE}.\mathtt{KeyPairGen}()$

$\mathbf{if}\ Keys[epk] = \bot$

$\quad k \leftarrow \mathcal{KE}.\mathtt{Combine}(sk, epk)$

$\quad \mathbf{if}\ (b = 0)$

$\quad\quad k \xleftarrow{\$} \{0,1\}^{|k|}$

$\quad \mathbf{fi}$

$\quad Keys[epk] \leftarrow k$

$\mathbf{fi}$

$\mathbf{return}\ (epk, Keys[epk])$

Key secrecy for $\mathcal{KE}$: success measure

$$\mathbf{Adv}_{\mathcal{KE}}^{\mathrm{MODH}}(t, q_{gen}, q_{com})$$

maximal advantage $\mathbf{Adv}_{\mathcal{KE}}^{\mathrm{MODH}}(\mathcal{A})$, where the maximum is taken over the adversaries $\mathcal{A}$ whose time complexity is at most t, making at most q_{gen} queries to $\mathcal{O}_{\mathrm{kgen}}^{b}$, q_{com} queries to $\mathcal{O}_{\mathrm{comb}}$ oracles.

Table of Contents

Multi-user setting is reducible

Proposition

$$\mathbf{Adv}_{\mathcal{AE}}^{\text{LOR-CCA}}(t, Q_e, Q_d, L_e, L_d, M_e, M_d; D) \leq$$
$$\leq D \cdot \mathbf{Adv}_{\mathcal{AE}}^{\text{LOR-CCA}}(t + T, q_e, q_d, l_e, l_d, \mu_e, \mu_d; 1),$$

- $T = D + \sum_{i=1}^{D} (Q_e[i] + Q_d[i] + L_e[i] + L_d[i])$,
- $q_x = \max_{1 \leq i \leq D} Q_x[i]$, $l_x = \max_{1 \leq i \leq D} L_x[i]$, $\mu_x = \max_{1 \leq i \leq D} M_x[i]$, $x \in \{e, d\}$.

Multi-user setting is reducible-2

Proposition

$$\mathbf{Adv}^{\text{INT-CTXT}}_{\mathcal{AE}}(t, Q_e, Q_v, L_e, L_v, M_e, M_v; D) \leq$$
$$\leq D \cdot \mathbf{Adv}^{\text{INT-CTXT}}_{\mathcal{AE}}(t + T, q_e, q_v, l_e, l_v, \mu_e, \mu_v; 1),$$

- $T = D + \sum_{i=1}^{D} (Q_e[i] + Q_v[i] + L_e[i] + L_v[i])$,
- $q_x = \max_{1 \leq i \leq D} Q_x[i]$, $l_x = \max_{1 \leq i \leq D} L_x[i]$, $\mu_x = \max_{1 \leq i \leq D} M_x[i]$, $x \in \{e, v\}$.

Multi-user setting is reducible: ideas

- Main idea: hybrid argument (keys k_i are independent)...

Multi-user setting is reducible: ideas

- Main idea: hybrid argument (keys k_i are independent)...
- i.e., choose one index j, on which oracle queries are redirected; model the others.

Multi-user setting is reducible: ideas

- Main idea: hybrid argument (keys k_i are independent)...
- i.e., choose one index j, on which oracle queries are redirected; model the others.
- We assume that key generation and processing one block of a text requires 1 unit of time.

MODH model is reducible

Proposition

Assume that the distribution of ephemeral public keys epk generated by $\mathcal{KE}$.`KeyPairGen` *is uniformly random on $EpkSet$. Then the following inequality holds:*

$$\mathbf{Adv}^{\mathrm{MODH}}_{\mathcal{KE}}(t, q_{gen}, q_{com}) \leq q_{gen} \cdot \mathbf{Adv}^{\mathrm{MODH}}_{\mathcal{KE}}(t + q_{gen} + q_{com}, 1, q_{com}) + \frac{2\, q_{gen}\, q_{com}}{|EpkSet|},$$

MODH model is reducible-2

- Main idea: again hybrid argument…

MODH model is reducible-2

- Main idea: again hybrid argument...
- but: might be some problem if the key epk generated inside $\mathcal{O}^{b}_{\mathrm{kgen}}$ collides with one of the keys epk queried by $\mathcal{A}$ to $\mathcal{O}_{\mathrm{comb}}$ oracle.

MODH model is reducible-2

- Main idea: again hybrid argument...
- but: might be some problem if the key $\boldsymbol{epk}$ generated inside $\mathcal{O}^b_{\mathrm{kgen}}$ collides with one of the keys $\boldsymbol{epk}$ queried by $\mathcal{A}$ to $\mathcal{O}_{\mathrm{comb}}$ oracle.
- Exclude this (bad) event: $\frac{q_{gen}\, q_{com}}{|EpkSet|}$ summand.

Results for ECIES

Assume that the distribution of ephemeral public keys epk generated by $\mathcal{KE}.\mathtt{KeyPairGen}$ is uniformly random on $EpkSet$.

Assume that the distribution of ephemeral public keys epk generated by $\mathcal{KE}$.**KeyPairGen** is uniformly random on $EpkSet$.

Proposition

$$\mathbf{Adv}^{\text{LOR-CCA}}_{\text{ECIES}}(t, q_e, q_d, l_e, l_d, \mu_e, \mu_d) \leq$$

$$\leq 2 \cdot \mathbf{Adv}^{\text{MODH}}_{\mathcal{KE}}(t + T_1, q_e, q_d) + q_e \cdot \mathbf{Adv}^{\text{LOR-CCA}}_{\mathcal{AE}}(t + T_2, q_e, q_d, l_e, l_d, \mu_e, \mu_d; 1) + \frac{q_e \cdot q_d}{|EpkSet|},$$

where $T_1 = q_e + q_d + l_e + l_d$, $T_2 = q_d + l_d + q_e(q_e + q_d + l_e + l_d + 2)$.

Results for ECIES - 2

Proposition

$$\mathbf{Adv}^{\text{INT-CTXT}}_{\text{ECIES}}(t, q_e, q_v, l_e, l_v, \mu_e, \mu_v) \leq$$
$$\leq \mathbf{Adv}^{\text{MODH}}_{\mathcal{KE}}(t + T_1, q_e, q_v) + q_e \mathbf{Adv}^{\text{INT-CTXT}}_{\mathcal{AE}}(t + T_2, q_e, q_v, l_e, l_v, \mu_e, \mu_v; 1) + \frac{q_e \cdot q_v}{|EpkSet|},$$

where $T_1 = q_e + q_v + l_e + l_v$, $T_2 = D + q_v + l_v + q_e \cdot (1 + q_e + q_v + l_e + l_v)$.

Results for ECIES: conclusions

- Main result: decompose the security of ECIES to the security of $\mathcal{AE}$ and $\mathcal{KE}$.

Results for ECIES: conclusions

- Main result: decompose the security of ECIES to the security of $\mathcal{AE}$ and $\mathcal{KE}$.
- One can instantiate ECIES with concrete $\mathcal{AE}$ and $\mathcal{KE}$ and obtain concrete estimates.

Results for ECIES: conclusions

- Main result: decompose the security of ECIES to the security of $\mathcal{AE}$ and $\mathcal{KE}$.
- One can instantiate ECIES with concrete $\mathcal{AE}$ and $\mathcal{KE}$ and obtain concrete estimates.
- Obtaining estimates for the (in)security of $\mathcal{AE}$ on a single key in $\mathrm{LOR\text{-}CCA}$ and $\mathrm{INT\text{-}CTXT}$ is a well-known problem; many results for specific schemes.

Results for ECIES: conclusions

- Main result: decompose the security of ECIES to the security of $\mathcal{AE}$ and $\mathcal{KE}$.
- One can instantiate ECIES with concrete $\mathcal{AE}$ and $\mathcal{KE}$ and obtain concrete estimates.
- Obtaining estimates for the (in)security of $\mathcal{AE}$ on a single key in $\mathrm{LOR\text{-}CCA}$ and $\mathrm{INT\text{-}CTXT}$ is a well-known problem; many results for specific schemes.
- $\mathcal{KE}$ in MODH is more elaborate...

$\mathcal{KE}$ in MODH: why problematic?

- Example: VKO scheme.

---

[9] Abdalla, Bellare, and Rogaway, “The oracle Diffie-Hellman assumptions and an analysis of DHIES.”

[10] Smart, “The exact security of ECIES in the generic group model.”

$\mathcal{KE}$ in MODH: why problematic?

- Example: VKO scheme.
- To estimate security we must take into consideration how hash function and group operation are interwined.

[9] Abdalla, Bellare, and Rogaway, "The oracle Diffie-Hellman assumptions and an analysis of DHIES."
[10] Smart, "The exact security of ECIES in the generic group model."

$\mathcal{KE}$ in MODH: why problematic?

- Example: VKO scheme.
- To estimate security we must take into consideration how hash function and group operation are interwined.
- “Bad interaction” may lead to the situation when DDH problem is hard, but ODH problem is easy.

[9] Abdalla, Bellare, and Rogaway, “The oracle Diffie-Hellman assumptions and an analysis of DHIES.”
[10] Smart, “The exact security of ECIES in the generic group model.”

$\mathcal{KE}$ in MODH: why problematic?

- Example: VKO scheme.
- To estimate security we must take into consideration how hash function and group operation are interwined.
- “Bad interaction” may lead to the situation when DDH problem is hard, but ODH problem is easy.
- Various “idealized” versions of the problem can be studied: Hash as a Random Oracle[9], Generic Group Model[10], etc.

---

[9] Abdalla, Bellare, and Rogaway, “The oracle Diffie-Hellman assumptions and an analysis of DHIES.”

[10] Smart, “The exact security of ECIES in the generic group model.”

- Abdalla, Michel, Mihir Bellare, and Phillip Rogaway. “The oracle Diffie-Hellman assumptions and an analysis of DHIES.” In: *Topics in Cryptology—CT-RSA 2001: The Cryptographers’ Track at RSA Conference 2001 San Francisco, CA, USA, April 8–12, 2001 Proceedings*. Springer. 2001, pp. 143–158.
- Akhmetzyanova, Liliya et al. “Security of Multilinear Galois Mode (MGM).” In: (2019). `https://eprint.iacr.org/2019/123`. URL: `https://eprint.iacr.org/2019/123`.
- Alekseev, Evgeny Konstantinovich et al. “On the cryptographic properties of algorithms accompanying the applications of standards GOST R 34.11-2012 and GOST R 34.10-2012.” In: *Matematicheskie Voprosy Kriptografii [Mathematical Aspects of Cryptography]* 7.1 (2016). In Russian, pp. 5–38.

Bellare, Mihir and Chanathip Namprempre. “Authenticated encryption: Relations among notions and analysis of the generic composition paradigm.” In: *Advances in Cryptology—ASIACRYPT 2000: 6th International Conference on the Theory and Application of Cryptology and Information Security Kyoto, Japan, December 3–7, 2000 Proceedings 6*. Springer. 2000, pp. 531–545.

Gayoso Martínez, V, L Hernández Encinas, and A Queiruga Dios. “Security and practical considerations when implementing the elliptic curve integrated encryption scheme.” In: *Cryptologia* 39.3 (2015), pp. 244–269.

Martínez, V Gayoso, L Hernández Encinas, et al. “A comparison of the standardized versions of ECIES.” In: *2010 Sixth International Conference on Information Assurance and Security*. IEEE. 2010, pp. 1–4.

Nozdrunov, Vladislav. “Parallel and double block cipher mode of operation (PD–mode) for authenticated encryption.” In: *TC 26* (2017).

Shoup, Victor. *A Proposal for an ISO Standard for Public Key Encryption*. Cryptology ePrint Archive, Paper 2001/112. **https://eprint.iacr.org/2001/112**. 2001. URL: **https://eprint.iacr.org/2001/112**.

Smart, Nigel P. "The exact security of ECIES in the generic group model." In: *Cryptography and Coding: 8th IMA International Conference Cirencester, UK, December 17–19, 2001 Proceedings 8*. Springer. 2001, pp. 73–84.

Thank you for your attention!

Author(s):

Tsaregorodtsev Kirill

Researcher at Cryptography laboratory,
JSRPC “Kryptonite”, Moscow, Russia
k.tsaregorodtsev@kryptonite.ru