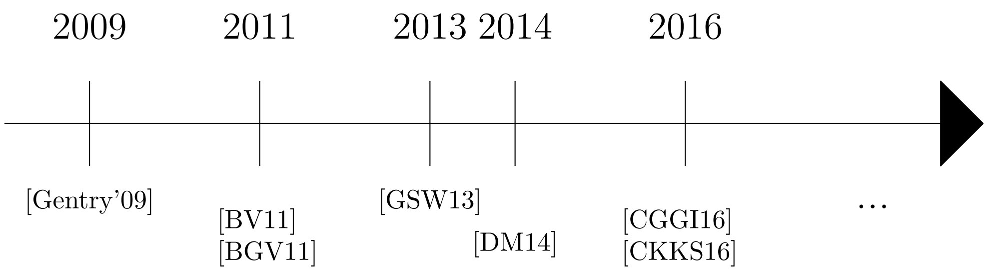
Forschungsinstitut
Cyber Defence
Universität der Bundeswehr München

FHE Beyond IND-CCA1 Security

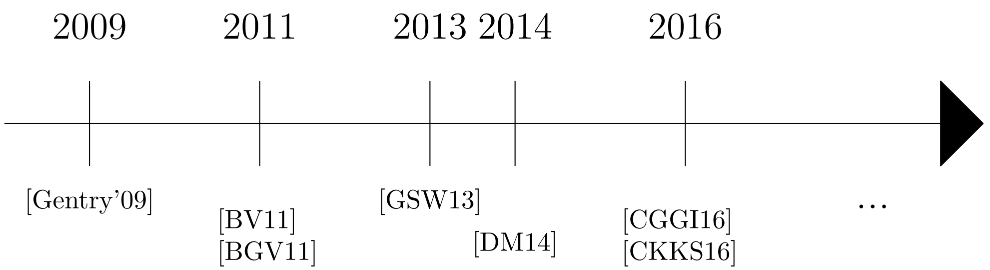
Mark Manulis and Jérôme Nguyen
Universität der Bundeswehr, München

Eurocrypt 2024

Selected history of FHE : Early days

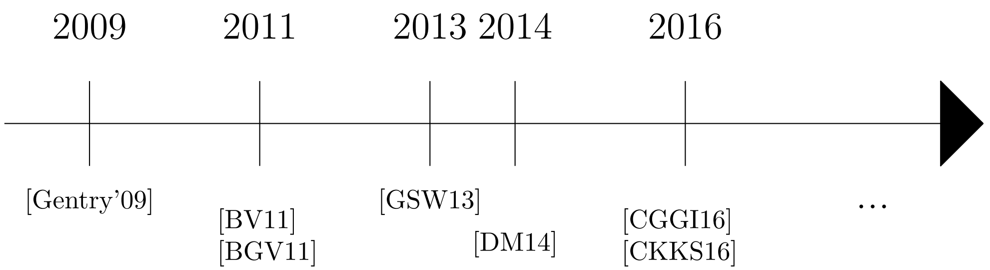


Selected history of FHE : Early days



- Research mainly focused on functionalities/efficiency
- Security guarantees: Chosen-plaintext attack (CPA)

Selected history of FHE : Early days



- Research mainly focused on functionalities/efficiency
- Security guarantees: Chosen-plaintext attack (CPA)

Reasonable start:

- FHE was too inefficient to be used in practice
- Applications seemed fine without stronger notions
- (FHE seems inherently vulnerable to chosen-ciphertext attacks)

Is CPA enough in practice?

Is CPA enough in practice?

No!

Is CPA enough in practice?

No!

(It depends)

Is CPA enough in practice?

No!

(It depends)

- Attacks on approximate FHE (CPA^{D})[LM21]

Is CPA enough in practice?

No!

(It depends)

- Attacks on approximate FHE (CPA^D)[LM21]
- Ciphertext validity attacks (CCVA) [CGG16]

Is CPA enough in practice?

No!

(It depends)

- Attacks on approximate FHE (CPA^D)[LM21]
- Ciphertext validity attacks (CCVA) [CGG16]
- Client-aided outsourcing attacks (FuncCPA) [AGHV22]

Is CPA enough in practice?

No!

(It depends)

- Attacks on approximate FHE (CPA^D)[LM21]
- Ciphertext validity attacks (CCVA) [CGG16]
- Client-aided outsourcing attacks (FuncCPA) [AGHV22]
- ...

Is CPA enough in practice?

No!

(It depends)

- Attacks on approximate FHE (CPA^D)[LM21]
- Ciphertext validity attacks (CCVA) [CGG16]
- Client-aided outsourcing attacks (FuncCPA) [AGHV22]
- …

What security notion should we aim for?

For “regular” encryption schemes : CCA2

$\mathcal{C}$hallenger $\qquad$ $\mathcal{A}$dversary

1st phase

$\xrightarrow{\text{pk}}$

$c_i = \text{Enc}_{pk}(x_i)$

$\text{Dec}_{sk}(\cdot)$

x_i

$\xleftarrow{(m_0, m_1)}$

Challenge phase

$\xrightarrow{c^* = \text{Enc}_{pk}(m_b)}$

2nd phase

$c_{i'} = \text{Enc}_{pk}(x_{i'})$

$\text{Dec}_{sk}(\cdot)$ $\quad$ If $c_{i'} \neq c^*$

$x_{i'}$

$\xleftarrow{\hat{b}}$

$b \stackrel{?}{=} \hat{b}$

CCA2: Impossible!

- FHE cannot be CCA2

$$
\begin{tikzcd}
\mathrm{Enc}_{pk}(m_b) \arrow[d] & & \mathrm{Enc}_{pk}(0) \arrow[d] & & \\
c^* & \oplus & c_0 \arrow[r] & c' \arrow[r] & \mathrm{Dec}_{sk} \arrow[d] \\
& & & & m_b
\end{tikzcd}
$$

Security for FHE schemes : CCA1

$\mathcal{C}$hallenger $\qquad\qquad$ $\mathcal{A}$dversary

1st phase

$\xrightarrow{\text{pk}}$

$c_i = \text{Enc}_{pk}(x_i)$ $\longrightarrow$ $\text{Dec}_{sk}(\cdot)$ $\xrightarrow{x_i}$

$\xleftarrow{(m_0, m_1)}$

Challenge phase

$c^* = \text{Enc}_{pk}(m_b)$ $\longrightarrow$

2nd phase

$\xleftarrow{\hat{b}}$

$b \stackrel{?}{=} \hat{b}$

CCA1: Impossible in practice (?)

CCA1: Impossible in practice (?)

- Positive results for general FHE (or leveled) [BSW12, CRRV17, YKT18]

CCA1: Impossible in practice (?)

- Positive results for general FHE (or leveled) [BSW12, CRRV17, YKT18]
- (Folklore) Bootstrapping schemes cannot be CCA1

CCA1: Impossible in practice (?)

- Positive results for general FHE (or leveled) [BSW12, CRRV17, YKT18]
- (Folklore) Bootstrapping schemes cannot be CCA1

Bootstrapping key is public

CCA1: Impossible in practice (?)

- Positive results for general FHE (or leveled) [BSW12, CRRV17, YKT18]

- (Folklore) Bootstrapping schemes cannot be CCA1

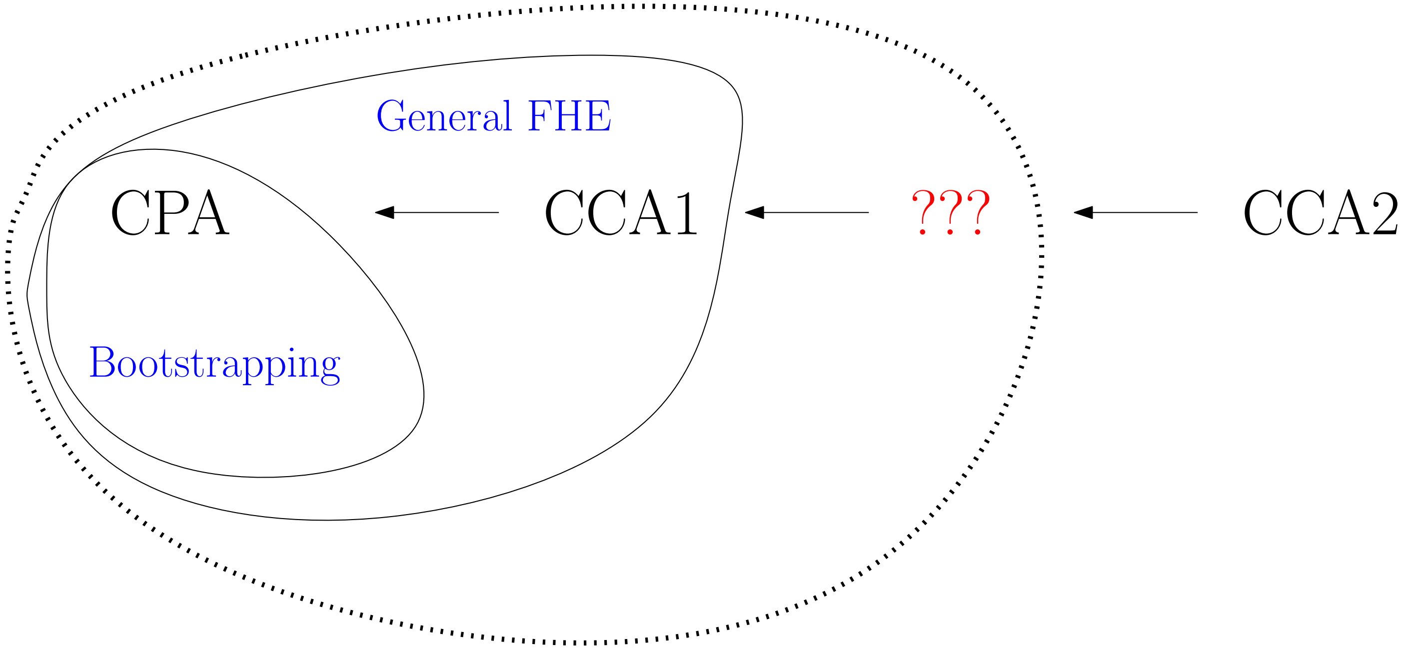
Bootstrapping key is public

$$\mathrm{Enc}_{pk}(sk) \downarrow \mathrm{bk}$$

$$\mathrm{bk} \longrightarrow \mathrm{Dec}_{sk} \downarrow sk$$

Questions

- Is it possible to relax CCA2 for FHE?
- Can bootstrapping schemes be stronger than CPA?



Our results

- Define a new security notion: IND-vCCA
  - Strictly between CCA1 & CCA2
  - Strongest among the (known) achievable notions for FHE
  - Equivalent formulation as “non-malleability definition”: TNM-vCCA

- Achievable in the ROM for (bootstrapping-based) FHE schemes from:
  - Passively secure FHE (CPA/CPA^D)
  - General CCA2 transformation
  - Succinct non-interactive argument of knowledge (SNARK)

The extraction algorithm (informal)

$$\mathrm{Eval}(f, c_1 \ldots, c_l) \to \hat{c}$$

The extraction algorithm (informal)

$$\mathrm{Eval}(f, c_1 \ldots, c_l) \rightarrow \hat{c}$$

Define Extract such that:

The extraction algorithm (informal)

$$\mathrm{Eval}(f, c_1 \ldots, c_l) \rightarrow \hat{c}$$

Define Extract such that:

$$\mathrm{Extract}(\hat{c}) \rightarrow (f, c_1, \ldots, c_l)$$

The extraction algorithm (informal)

$$\mathrm{Eval}(f, c_1 \ldots, c_l) \to \hat{c}$$

Define Extract such that:

$$\mathrm{Extract}(\hat{c}) \to (f, c_1, \ldots, c_l)$$

- $(c_1, \ldots, c_l)$ are fresh ciphertexts

The extraction algorithm (informal)

$$\mathrm{Eval}(f, c_1 \ldots, c_l) \rightarrow \hat{c}$$

Define Extract such that:

$$\mathrm{Extract}(\hat{c}) \rightarrow (f, c_1, \ldots, c_l)$$

- $(c_1, \ldots, c_l)$ are fresh ciphertexts
- Soundness:

The extraction algorithm (informal)

$$\mathrm{Eval}(f, c_1 \ldots, c_l) \to \hat{c}$$

Define Extract such that:

$$\mathrm{Extract}(\hat{c}) \to (f, c_1, \ldots, c_l)$$

- $(c_1, \ldots, c_l)$ are fresh ciphertexts
- Soundness:

$$\mathrm{Dec}_{sk}(\hat{c}) = f(\mathrm{Dec}_{sk}(c_1), \ldots, \mathrm{Dec}_{sk}(c_l))$$

The vCCA oracle

Define a new oracle using the extraction algorithm:

$$\mathcal{O}_{\text{vCCA}}(\hat{c}):$$

$$1)\ \text{Extract}(\hat{c}) \rightarrow (f, c_1, \ldots, c_l)$$

$$2)\ \text{If}\quad c^* \notin (c_1, \ldots, c_l)$$

$$\text{Return}\ \ \text{Dec}_{sk}(\hat{c})$$

IND-vCCA

$\mathcal{C}$hallenger $\qquad$ $\mathcal{A}$dversary

$c_i = \mathrm{Enc}_{pk}(x_i)$

pk

1st phase $\qquad$ $\mathrm{Dec}_{sk}(\cdot)$

(m_0, m_1) $\qquad$ x_i

Challenge phase

$c^* = \mathrm{Enc}_{pk}(m_b)$

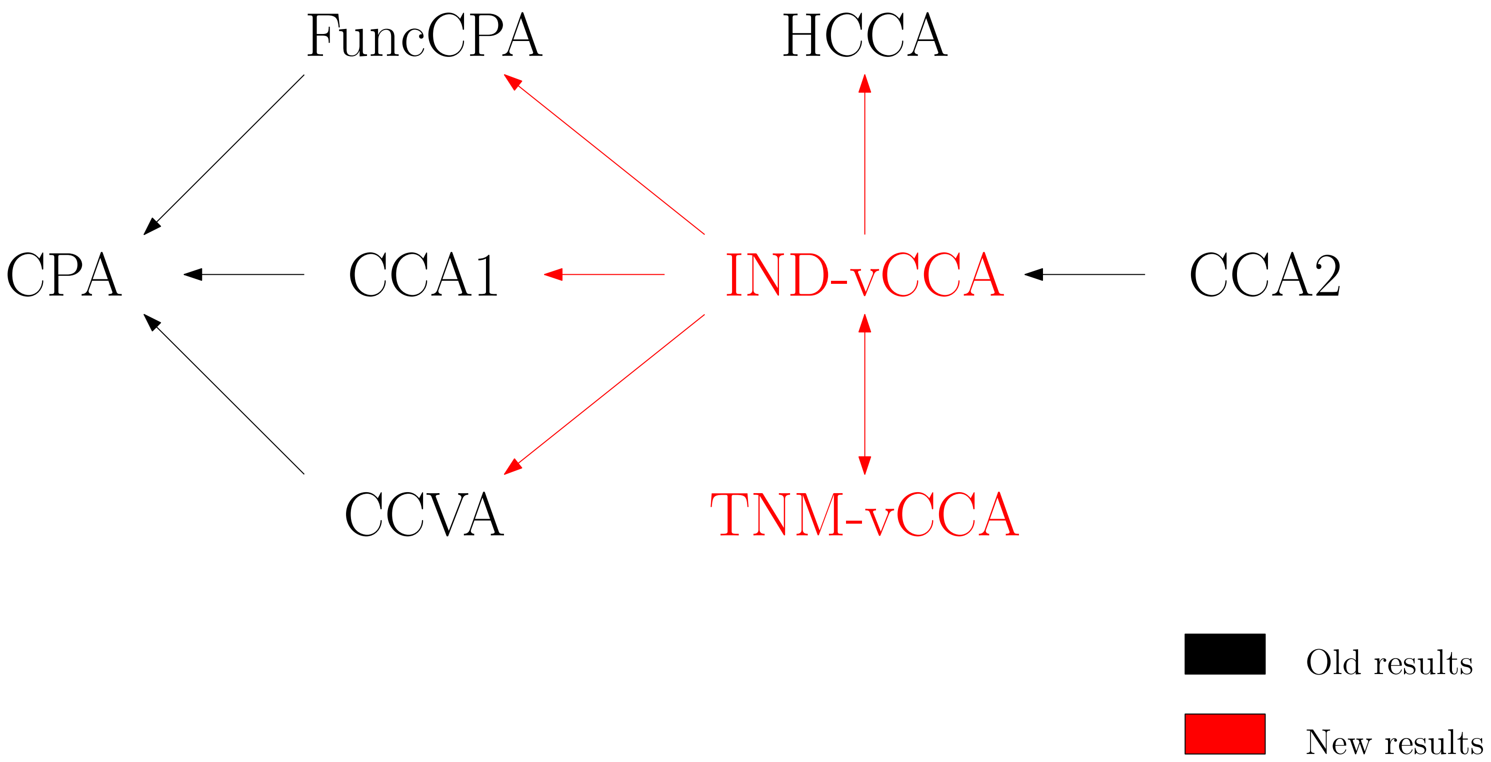
2nd phase $\qquad$ $\hat{c}$

$\boxed{\mathcal{O}_{\text{vCCA}}(\cdot)}$

$\hat{b}$ $\qquad$ $x_{i'}$

$b \stackrel{?}{=} \hat{b}$

vCCA: Simplified relationship graph



Achieving vCCA: The CCA2 transform

- First step: Lose the homomorphism!
  - Embed the FHE scheme in a CCA2 encryption scheme

Achieving vCCA: The CCA2 transform

- First step: Lose the homomorphism!
  - Embed the FHE scheme in a CCA2 encryption scheme

- Let CCA2(m) be a generic CCA2 transform such that:

Achieving vCCA: The CCA2 transform

- First step: Lose the homomorphism!

  Embed the FHE scheme in a CCA2 encryption scheme

- Let CCA2(m) be a generic CCA2 transform such that:

$$\mathrm{CCA2}(m) \to c = (c', t)$$

With

$$\mathrm{Enc}_{pk}(m) \longrightarrow c'$$

Achieving vCCA: Example CCA2 transform

Symmetric FHE

- Encrypt-then-MAC:

(c’, MAC(c’))

- Encrypt-then-Sign:

(c’, Sign(c’))

Asymmetric FHE

- Naor-Yung: Double encryption & NIZK

$(c_1, c_2, \mathrm{Proof}(c_1,c_2,m))$

- Fujisaki-Okamoto

Achieving vCCA: Handling bootstrapping

If the scheme uses a bootstrapping key bk = Enc(sk):

- Do NOT release CCA2(sk)
- The bootstrapping key remains bk – not a valid CCA2 ciphertext

Achieving vCCA: The SNARK

Use SNARK to prove:

- Computation of the evaluation algorithm
- Knowledge of corresponding valid CCA2 ciphertexts

$$\text{CCA2}$$

$$c_1 = (c'_1, t_1)$$

$$\vdots$$

$$c_l = (c'_l, t_l)$$

$$\xleftrightarrow{\text{Prove}}$$

$$\text{Prove}$$

$$\text{Eval}(f, c'_1, \ldots, c'_l) = \hat{c}$$

Achieving vCCA: The SNARK properties

The SNARK must be:

- Non-rewinding
- Simulation-extractable (non-malleable)
- Black-box

Suitable SNARKs exist in the random oracle model

Achieving vCCA: Putting it together

$$
\begin{tikzcd}
\mathrm{Enc}_{pk}(m_1) \to c_1 \arrow[dr] & & \\
\vdots \arrow[r] & \mathrm{Eval}(f, c_1 \ldots, c_l) \to \hat{c} \arrow[r] & \mathrm{Dec}_{sk}(\hat{c}) \to f(m_1, \ldots, m_l) \\
\mathrm{Enc}_{pk}(m_1) \to c_l \arrow[ur] & &
\end{tikzcd}
$$

Achieving vCCA: Putting it together

$$\mathrm{Enc}_{pk}(m_1) \to c_1$$

$$\vdots$$

$$\mathrm{Enc}_{pk}(m_l) \to c_l$$

$$\mathrm{Eval}(f, c_1 \ldots, c_l) \to \hat{c} \longrightarrow \mathrm{Dec}_{sk}(\hat{c}) \to Attackvector$$

Achieving vCCA: Putting it together

$$\begin{tikzcd}
\mathrm{CCA2}_{pk}(m_1) \to c_1 = (c'_1, t_1) \arrow[dr] & & \\
\vdots \arrow[r] & \mathrm{Eval}(f, c'_1 \ldots, c'_l) \to \hat{c} \arrow[r] & \mathrm{Dec}_{sk}(\hat{c}) \to \mathit{Attackvector} \\
\mathrm{CCA2}_{pk}(m_l) \to c_l = (c'_l, t_l) \arrow[ur] & &
\end{tikzcd}$$

Achieving vCCA: Putting it together

$$
\begin{tikzcd}
\mathrm{CCA2}_{pk}(m_1) \to c_1 = (c'_1, t_1) \arrow[dr] \arrow[drr, bend left, "\text{Prove}"] & & & \\
\vdots \arrow[r] & \mathrm{Eval}(f, c'_1 \ldots, c'_l) \to & \hat{c} \arrow[r] & \mathrm{Dec}_{sk}(\hat{c}) \to f(m_1, \ldots, m_l) \\
\mathrm{CCA2}_{pk}(m_l) \to c_l = (c'_l, t_l) \arrow[ur] & & &
\end{tikzcd}
$$

Achieving vCCA: Putting it together

$$\begin{tikzcd}[column sep=small]
\mathrm{CCA2}_{pk}(m_1) \to c_1 = (c'_1, t_1) \arrow[dr] \arrow[drr, bend left, "\mathrm{Prove}"] & & \\
\vdots \arrow[r] & \mathrm{Eval}(f, c'_1 \ldots, c'_l) \to & \hat{c} \arrow[r] & \mathrm{Dec}_{sk}(\hat{c}) \to f(m_1, \ldots, m_l) \\
\mathrm{CCA2}_{pk}(m_l) \to c_l = (c'_l, t_l) \arrow[ur] & &
\end{tikzcd}$$

Theorem: This construction is IND-vCCA secure.

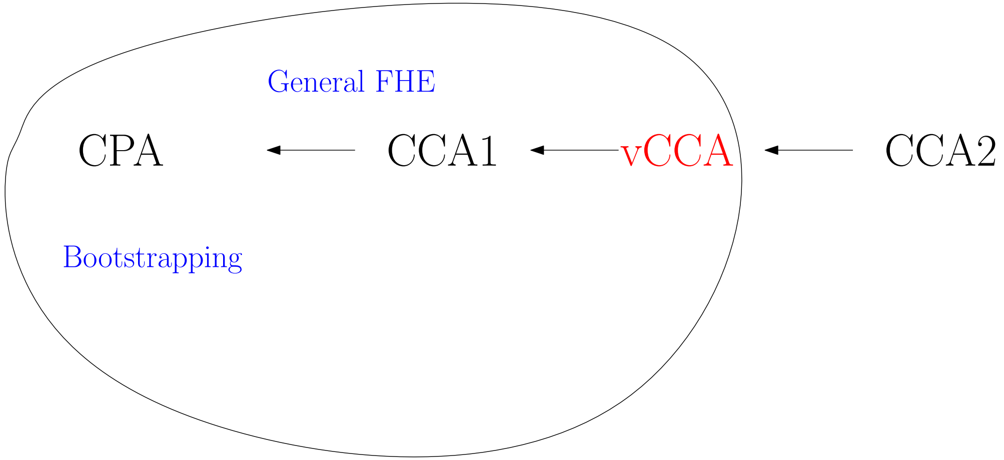
Proof Idea: Reduce to CCA2 security. Answer decryption queries by extracting each query.

Conclusion

Proposed new security notion for FHE schemes: IND-vCCA

It is:

- Achievable through generic transformation in the ROM
- The strongest achievable security notion known for FHE
- Allows for bootstrapping



Thank You!