National Cyber Security Centre

Security Goals for an Accordion Mode: Release of Unverified Plaintext and Multi-user security

Guy B

UK NCSC

June 2024

Our approach summarised

"Secure by design" philosophy
- Security should not depend on end-users being experts in cryptography
- High threat use cases require long-term security

Prioritise confidence over efficiency
- Robustness over performance
- Simplicity of security analysis

Formalising Two Security Goals

Release of Unverified Plaintext (RUP)

- Motivation
- A real-world RUP vulnerability
- Subtleties in security definitions
- NIST’s encode-then-encipher proposal achieves strong RUP security

Security Calculations

- Real-world advantage bounds must be usable in practice
- What parameters we can control
- Designing bounds to be instantiated

Application to MRAE

National Cyber
Security Centre

Release of Unverified Plaintext (RUP)

Release of Unverified Plaintext

RUP happens when failed decryption attempts are not fully discarded

Examples:
- Buffers containing putative plaintext not cleared
- Authentication checks omitted
- Compiler reorders authentication check with follow-on processing
- Implementation returns error codes/does padding checks
- Can lead to practical attacks, eg Efail [10]

Effect:
- Decryptor leaks the putative plaintext despite failure of verification
- Decryptor processes the putative plaintext despite failure of verification

Possible risks:
- Adversary learns secret information from this leak
- Adversarial control of putative plaintext influences actions of the decryptor

We view robustness against RUP as essential

RUP security games

Typical setup: give distinguisher access to third “decrypt leakage” oracle
In ideal case, $Leak$ is simulated by $\mathcal{S}$
In real case, $Leak$ chosen to model likely mis-implementation of decryption

$$vs?$$

$$Enc, Dec, Leak \qquad \qquad \$, \perp, \mathcal{S}$$

Defining $Leak$ can be complicated:

- Is “likely mis-implementation” well defined?
- Should we also allow for leakage of any variable-length buffer, as well as putative plaintext?

RUP security games – Further issues

$vs?$

$Enc, Dec, Leak$ $\quad$ $\$, \perp, \mathcal{S}$

Defining $\mathcal{S}$ is nuanced:

- Security notions PA1 [1], AE-RUP [6] allow $\mathcal{S}$ to use the transcript of queries to Enc
  - Too weak: implies leakage can contain information about plaintexts
  - E.g. GCM is PA1 secure, but would not block Efail

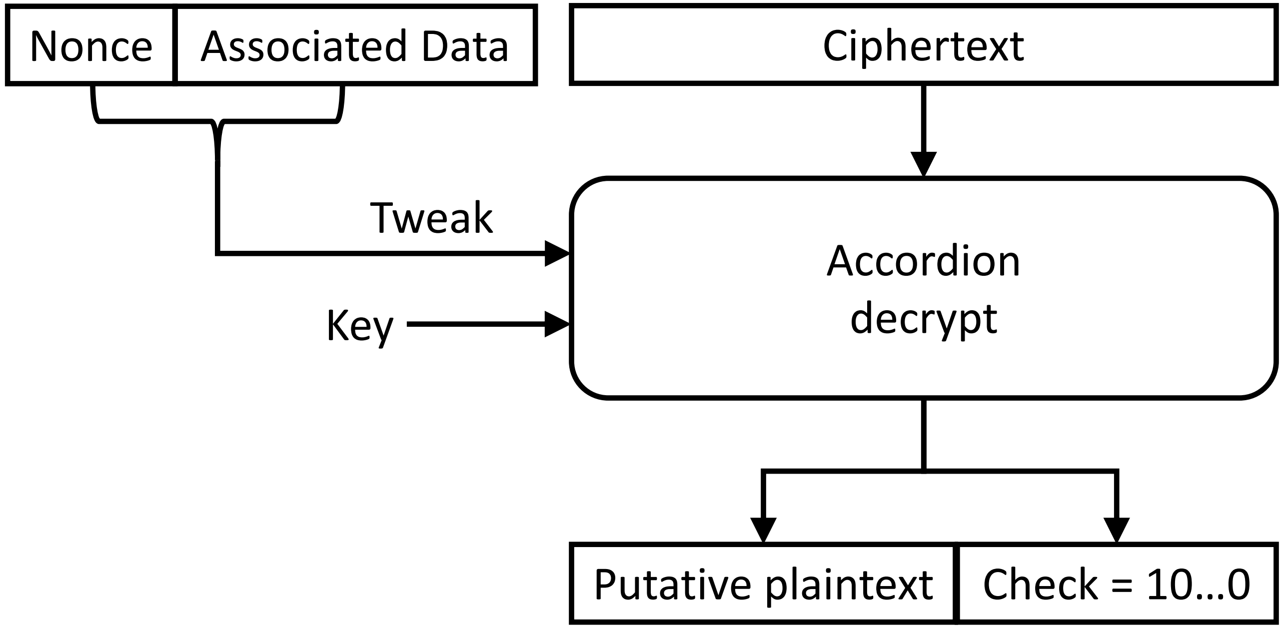
- Security notion SAE [3] is stronger - $\mathcal{S}$ may not view past transcript
  - Implies leakage cannot contain information about plaintexts
  - Still too weak: attacker can still exert control over leakage
  - E.g. an implementation that forwards the ciphertext on decryption failure is SAE secure, but undesirable

- On fresh inputs, $\mathcal{S}$ should output independent uniform random data
  - The proposal RUPAE [2] achieves this
  - We see strong RUP security - in this sense - as essential

RUP from Encode-then-Encipher

At birthday security levels, strong RUP security automatically follows from the proposed encode-then-encipher technique [7]

NCSC would like to see strong RUP security added as a design goal for the proposed accordion mode for AEAD



- Here the Accordion is a tweakable VIL-SPRP, so adversaries have no capability to learn from, or control, outputs to this function or its inverse

National Cyber Security Centre

Security Calculations

Security Bounds

A simple birthday advantage bound from a security proof might look like:

$$c\sigma^2/2^n$$

However, we deploy systems with many independent users, and wish to model adversaries attacking them all at once

Making a block cipher assumption, applying a standard hybrid argument for multi-user security (with per-user query restrictions), and requiring a security margin yields:

$$\mu c\sigma^2/2^n + \mu t/2^k \leq \varepsilon$$

Note that unlike some texts (e.g. [4]), we model each user as maintaining an independent query limit

Var.	Meaning
c	Small constant in proof
σ	Adversary query complexity budget
n	Block cipher block size
μ	Number of users (keys)
t	Adversary work budget
k	Block cipher key size
ε	Security margin

National Cyber Security Centre

Security budgets can be exceeded in large deployments

Taking the $\mu t/2^k$ term as negligible and rearranging gives:

$$\mu\sigma^2/\epsilon \leq 2^n$$

An example large deployment:

- $\mu \approx 2^{20}$ independent keys (users)
- each processing $\sigma \approx 2^{50}$ data
- for AES block size $n = 128$
- With proof constant $c \approx 16$
- Choice of confidence $\varepsilon \approx 10^{-9}$ as in NIST [8]

The inequality does not hold for large deployments

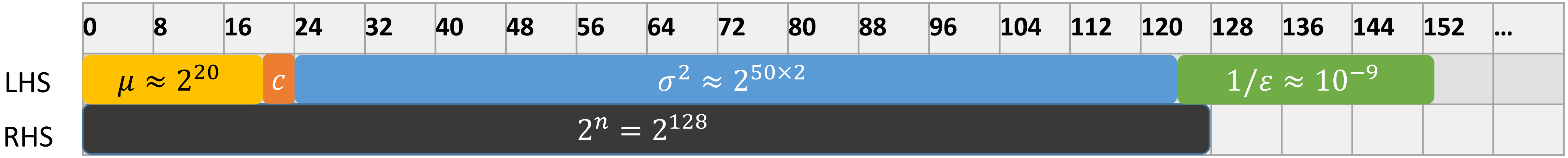


Figure: "Security budget", illustrates log-contribution of each term in the inequality

National Cyber Security Centre

Visualisations for different types of bound

Design	Visualization of inequality	Comments
Birthday bound with AES	μ, c, $\sigma^2 \approx 2^{50\times2}$, $1/\varepsilon$ $2^n = 2^{128}$	Not provably secure
Birthday bound with AES, restrictive per-user query limit	μ, c, $\sigma^2 \approx 2^{37\times2}$, $1/\varepsilon$ $2^n = 2^{128}$	Provably secure, but increasingly impractical to deploy
Beyond-birthday $\sigma^3/2^{2n}$ with AES	μ, c, $\sigma^3 \approx 2^{50\times3}$, $1/\varepsilon$ $2^{2n} = 2^{2\times128}$ …	Can be deployed now with AES as drop-in GCM replacement
Birthday mode, wider primitive $\sigma^2/2^N$, $N \gg 128$	μ, c, $\sigma^2 \approx 2^{50\times2}$, $1/\varepsilon$ *Larger Security budget* …	Desirable longer-term option

What variables can we control?

$$\mu\sigma^2/2^n + \mu t/2^k \leq \epsilon$$

Reconsider what terms make up this bound:

A more fine-grained bound can be helpful when instantiating, because we have more or less control over different variables

- Some we can easily enforce bounds
- Some we can estimate weak bounds from modelling
- Some are very hard to estimate

- Some common examples:

Var	Purpose	Restrictions
$\boldsymbol{\sigma_e}$	Total encrypted blocks per key	Each user can track and restrict independently
$\boldsymbol{\Sigma_e}$	Total encrypted blocks, all keys	Can’t enforce restriction without system-wide coordination
$\boldsymbol{\sigma_d}$	Total decrypted blocks per key	Can’t enforce a tight decryption limit (would enable DDOS attacks) can sometimes deduce soft limit from device bandwidth
$\boldsymbol{\mu}$	number of users/keys	Cannot enforce restriction without system-wide coordination Easier to estimate when keys are rotated on fixed schedule
$\boldsymbol{\epsilon}$	Security margin	Codifies user risk tolerance

National Cyber Security Centre

Visualisations of choices of variables (not to scale)

Design	Visualization of inequality	Comments
Generic Birthday bound	μ · c · σ · σ · $1/\varepsilon$ $2^n = 2^{128}$	For comparison, a σ^2 bound as before.
Representative of a tighter birthday bound	μ · c · q_e · $q_e + q_d \approx q_d$ · $1/\varepsilon$ $2^n = 2^{128}$	Even just splitting out q_e, q_d gets us closer to our goal
GCM Reconsidered [9, Thm 2]	μ · c · $q_e + q_d \approx q_d$ · $\sigma + q_e \approx \sigma$ · $1/\varepsilon$ $2^n = 2^{128}$	$\approx 64(q_e + q_d)(\sigma + q_e)/2^n$
VIGORNIAN first term [5, Thm 32]	μ · c · $q_e + 2q_d \approx 2q_d$ · $q_e + 2q_d \approx 2q_d$ · $q_e + 2q_d \approx 2q_d$ · $1/\varepsilon$ $2^{2n} = 2^{2\times128}$ …	$\approx 2^{10}\,\mu(q_e + 2q_d)^3/2^{2n}$

Interactions between MRAE and other security goals

Nonce repeats in decrypt queries

- Note that it is not feasible to prevent these
- In the event of RUP, security is bounded by the MRAE security level, even in a nonce-respecting mode

Limitation of MRAE security:

- Standard security analysis for an AEAD mode assesses against a TPRF
- By contrast, best attainable security while remaining decryptable is a TPRI [e.g. 7]
    - Tweakable Pseudo-Random Injection: distinct inputs on the same tweak give distinct outputs
- Security separation of TPRI and TPRF is birthday in the length of the output
    - distinction arises in the event of nonce repeats
- AEAD cannot generically attain TPRF security better than birthday in short message lengths

Should we adopt alternative idealisation of PRI?

- Potentially beyond-birthday secure even when misused
- Non-standard security notion likely not well understood by protocol designers, leading to fragility

Summary

Release of Unverified Plaintext (RUP)

- We hold “Secure by design” philosophy – security should not depend on end-users being experts in cryptography
- Subtleties in security definitions – we prefer a strong definition
- NIST’s encode-then-encipher proposal achieves this strong notion of RUP security
- We would like to see this added as a requirement

Security Calculations

- Real-world advantage bounds must be usable in practice
- Bounds should be constructed from parameters we can control

National Cyber Security Centre

Bibliography

[1] E. Andreeva, A. Bogdanov, A. Luykx, B. Mennink, N. Mouha, and K. Yasuda. How to securely release unverified plaintext in authenticated encryption. In P. Sarkar and T. Iwata, editors, ASIACRYPT 2014, Part I, volume 8873 of LNCS, pages 105–125. Springer, Heidelberg, Dec. 2014.
[2] T. Ashur, O. Dunkelman, and A. Luykx. Boosting authenticated encryption robustness with minimal modifications. In J. Katz and H. Shacham, editors, CRYPTO 2017, Part III, volume 10403 of LNCS, pages 3–33. Springer, Heidelberg, Aug. 2017.
[3] G. Barwell, D. Page, and M. Stam. Rogue decryption failures: Reconciling AE robustness notions. In J. Groth, editor, 15th IMA International Conference on Cryptography and Coding, volume 9496 of LNCS, pages 94–111. Springer, Heidelberg, Dec. 2015.
[4] M. Bellare, R. Canetti, and H. Krawczyk. Pseudorandom functions revisited: The cascade construction and its concrete security. In 37th FOCS, pages 514–523. IEEE Computer Society Press, Oct. 1996.
[5] P. Campbell. GLEVIAN and VIGORNIAN: Robust beyond-birthday AEAD modes. Cryptology ePrint Archive, Report 2023/1379, 2023. https://eprint.iacr.org/2023/1379.
[6] D. Chang, N. Datta, A. Dutta, B. Mennink, M. Nandi, S. Sanadhya, and F. Sibleyras. Release of unverified plaintext: Tight unified model and application to ANYDAE. IACR Trans. Symm. Cryptol., 2019(4):119–146, 2019.
[7] V. T. Hoang, T. Krovetz, and P. Rogaway. Robust authenticated-encryption AEZ and the problem that it solves. In E. Oswald and M. Fischlin, editors, EUROCRYPT 2015, Part I, volume 9056 of LNCS, pages 15–44. Springer, Heidelberg, Apr. 2015.
[8] N. Mouha and M. Dworkin. Report on the block cipher modes of operation in the NIST SP 800-38 series. NIST IR 8459 ipd, 2023.
[9] Y. Niwa, K. Ohashi, K. Minematsu, and T. Iwata. GCM security bounds reconsidered. In G. Leander, editor, FSE 2015, volume 9054 of LNCS, pages 385–407. Springer, Heidelberg, Mar. 2015.
[10] D. Poddebniak, C. Dresen, J. Müller, F. Ising, S. Schinzel, S. Friedberger, J. Somorovsky, and J. Schwenk. Efail: Breaking S/MIME and OpenPGP email encryption using exfiltration channels. In 27th USENIX Security Symposium (USENIX Security 18), pages 549–566, 2018.

With thanks to the Cryptobib effort for references