The Security of ChaCha20-Poly1305 in the Multi-User Setting

Jean Paul Degabriele **Jérôme Govinden** Felix Günther Kenneth G. Paterson

TECHNISCHE UNIVERSITÄT DARMSTADT

ETH zürich

ACM CCS 2021

Outline

Motivation

ChaCha20-Poly1305 Usage in Protocols

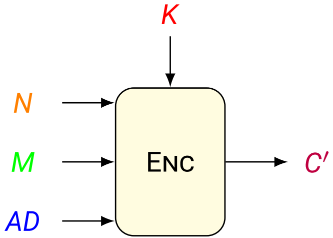
- ChaCha20-Poly1305 and AES-GCM (Galois Counter Mode) are the most popular AEAD schemes
- ChaCha20-Poly1305 is the default AEAD scheme in OpenSSH, WireGuard, OTRv4, and the Bitcoin Lightning Network
- ChaCha20-Poly1305 is recommended (after GCM) in TLS, DTLS and QUIC

$\rightarrow$ There is no correct security proof for ChaCha20-Poly1305
$\rightarrow$ We rectify the situation and obtain some surprising results

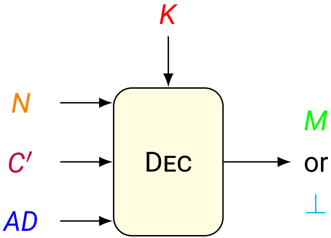
Background

Nonce-Based AEAD (Authenticated Encryption with Associated Data) Scheme
Syntax

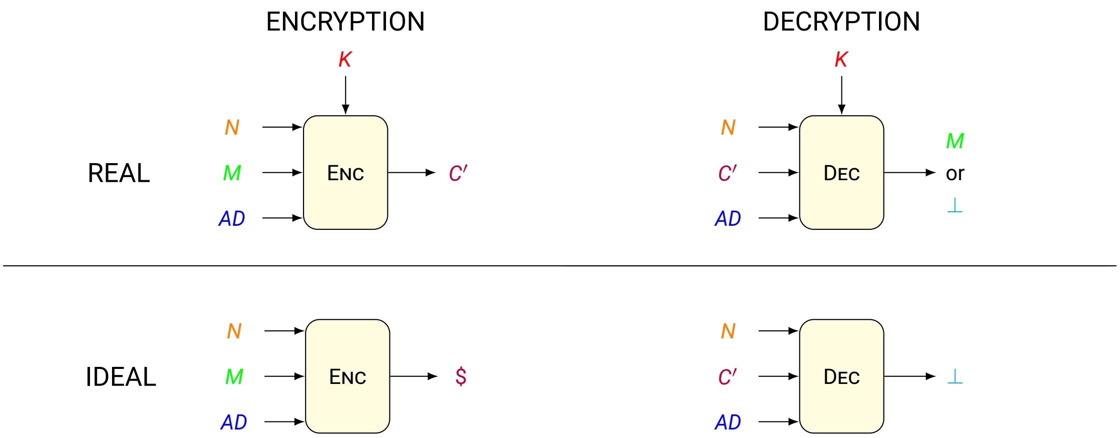
ENCRYPTION
(by sender)



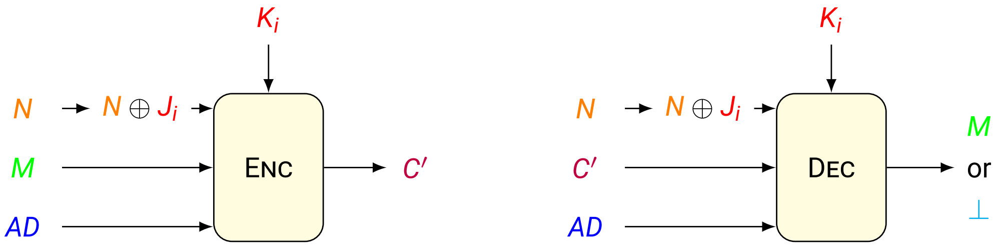
DECRYPTION
(by receiver)



Nonce-Based AEAD Security



Nonce-Randomized AEAD



- Used in widely deployed protocols such as TLS and QUIC
- Technique introduced in TLS 1.3 specification, intuitively, to mitigate multi-users attacks
- Formal justification obtained only later in [BT16; HTT18]

ChaCha20-Poly1305 Previous Security Analyses

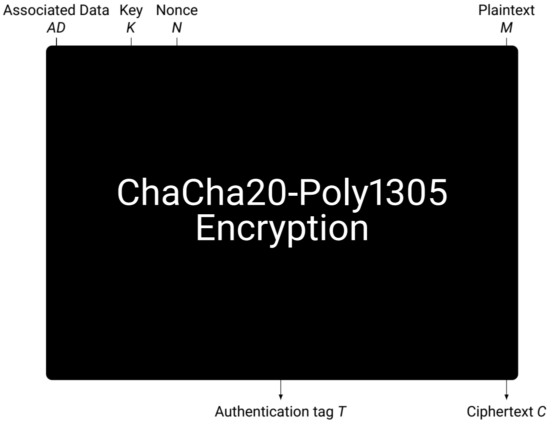
- ChaCha20 and Poly1305 were designed separately and independently by Bernstein
- They were combined into an AEAD scheme by Langley without security proof
- The only dedicated security analysis is in an unpublished note by Procter on IACR ePrint

Security analyses focuses mostly on AES-GCM:

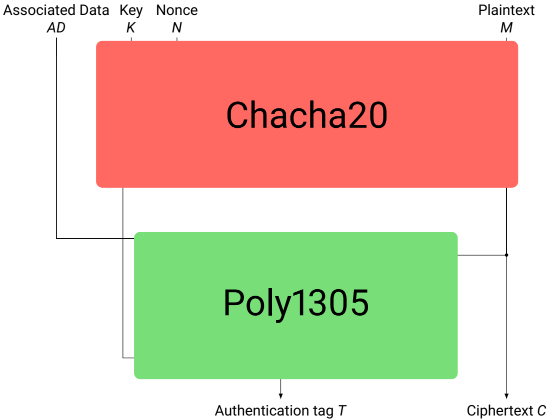
	AES-GCM	**ChaCha20-Poly1305**
Single-User:	[McGrew and Viega, *INDOCRYPT 2004*] [Iwata, Ohashi, and Minematsu, *CRYPTO 2012*] [Niwa et al., *FSE 2015*]	~~[Procter, IACR ePrint 2014]~~ The proof is incorrect
Multi-User:	[Bellare and Tackmann, *CRYPTO 2016, Part I*] [Luykx, Mennink, and Paterson, *ASIACRYPT 2017, Part II*] [Hoang, Tessaro, and Thiruvengadam, *ACM CCS 2018*]	~~Procter's Bound~~ ~~+~~ ~~Hybrid Argument~~

The Construction of ChaCha20-Poly1305

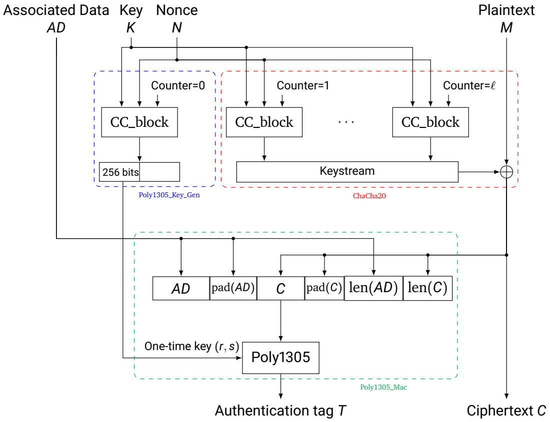
The ChaCha20-Poly1305 AEAD Scheme



The ChaCha20-Poly1305 AEAD Scheme

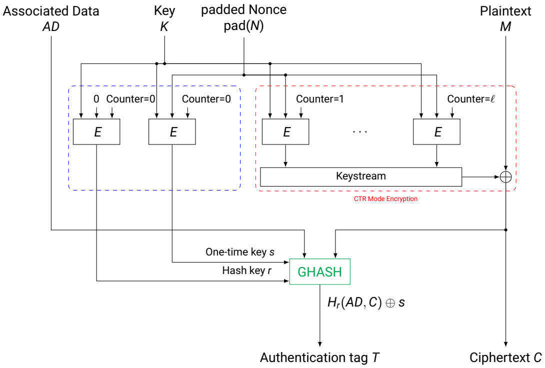


The ChaCha20-Poly1305 AEAD Scheme

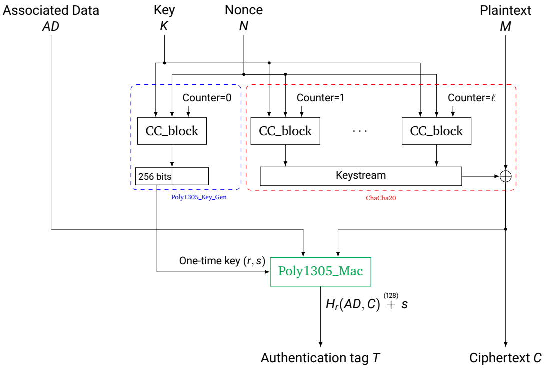


Differences between AES-GCM and ChaCha20-Poly1305

AES-GCM



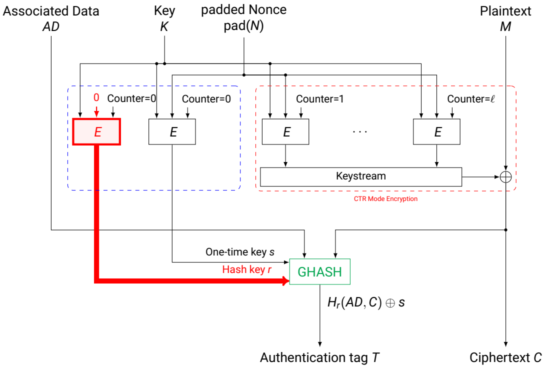
ChaCha20-Poly1305



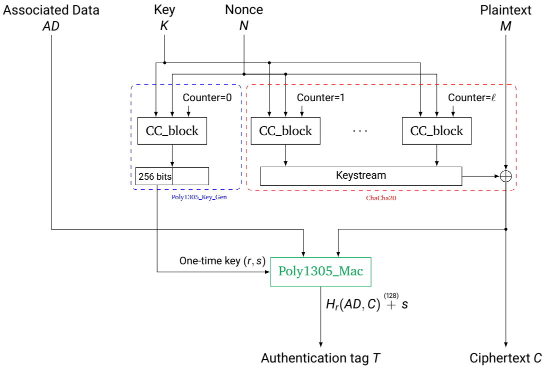
Differences between AES-GCM and ChaCha20-Poly1305

In the Construction (One-time Hash Key)

AES-GCM



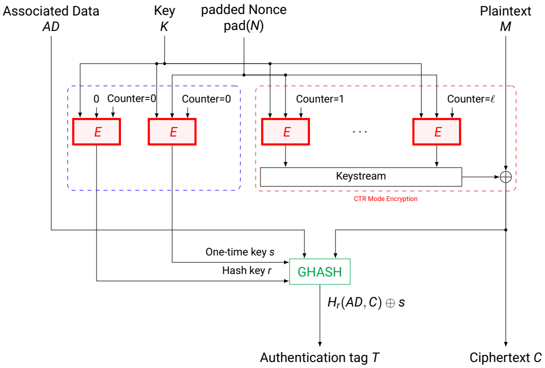
ChaCha20-Poly1305



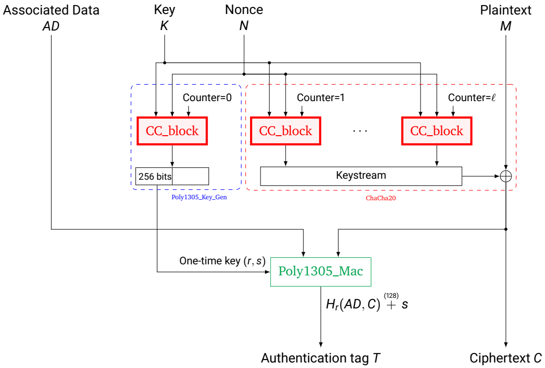
Differences between AES-GCM and ChaCha20-Poly1305

In the Blocks Generation

AES-GCM

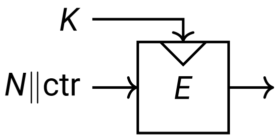


ChaCha20-Poly1305



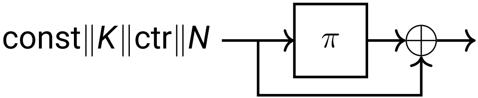
Differences between AES-GCM and ChaCha20-Poly1305
In the Blocks Generation

Block cipher in AES-GCM



- Block size: 128 bits
- AES

ChaCha20 block function



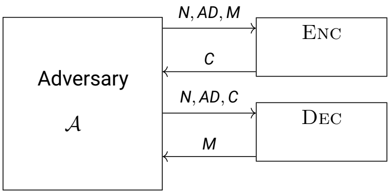
- Block size: 512 bits
- ARX construction

Security Analysis of ChaCha20-Poly1305

Previous Security Analysis

In the Single-User Setting

Single-User Setting



Procter's Bound

$$\mathsf{Adv}^{\mathrm{AE}}_{\mathrm{ChaCha20\text{-}Poly1305}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathrm{PRF}}_{\mathrm{CC\_block}}(\mathcal{A}_{\mathrm{prf}}) + \frac{3 \cdot q_v \cdot \ell_m}{2^{104}}$$

- In the Single-User Setting
  $\rightarrow$ does not consider multi-user attacks
- In the Standard Model: CC_block is assumed to be a PRF
  $\rightarrow$ does not explicitly quantify local computations advantage
- The original proof is incorrect

$\rightarrow$ We provide a new proof under the same assumption and recover the same security bound

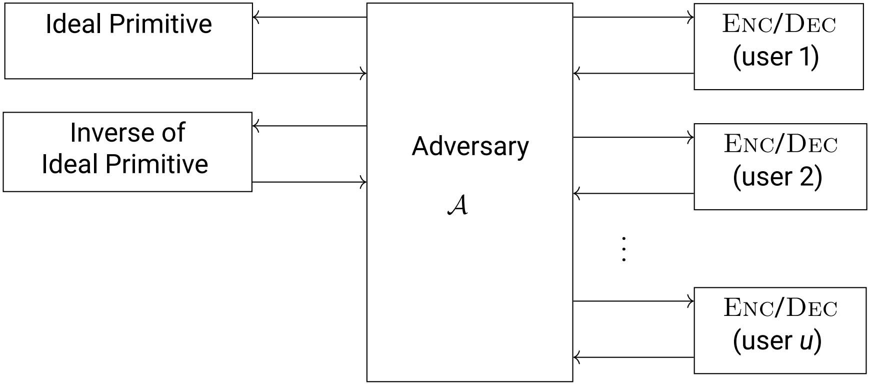
The Relevance of Multi-User Security

- It **better captures real world threats such as state-actors** that
  - Are able to **eavesdrop and collect en masse the data of multiple users** over the internet traffic,
  - Have **large computational resources**, which they can use for precomputation.

- It is the **preferred setting for choosing the parameters of many protocols**:
  - It is used to **determine rekeying frequencies** for AEAD in TLS, DTLS, and QUIC.
  - For protocols such as DTLS and QUIC, operating over UDP, it is used to **determine the number of failed decryption queries allowed** before terminating the session.

→ There is no Multi-User Security analysis for ChaCha20-Poly1305 available to practitioners.

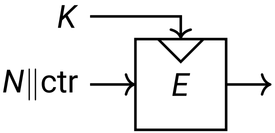
The Multi-User Security Model



Modelling the Underlying Primitive

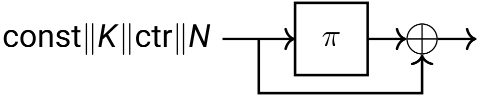
Ideal Cipher vs Ideal Permutation Model

Block cipher in AES-GCM



- For each key K, E_K is a different permutation
- No output collision for a single key K
- [BT16; HTT18] $\rightarrow$ E is an **ideal cipher**

ChaCha20 block function



- Uses only one permutation π
- Not indifferentiable from a random function
- In our proof $\rightarrow$ π is an **ideal permutation**

Proof Overview

- The proof is based on the H-coefficient technique
- It follows a similar structure as [HTT18] but with some noticeable differences:
  - It is done in a different model (i.e., ideal permutation model)
  - Good and Bad transcripts are defined differently $\rightarrow$ because block output collisions are possible
  - Some terms, corresponding to Bad transcripts, are bounded differently and improved

Multi-User Security Bound

$$\mathrm{Adv}^{\mathrm{muAE}}_{\mathrm{ChaCha20\text{-}Poly1305}[\pi]}(\mathcal{A}) \leq \underbrace{\frac{3 \cdot q_v(\ell_m + 1)}{2^{104}}}_{\text{Capture Forgery Attacks}} + \underbrace{\frac{p \cdot (d + 512)}{2^{256}}}_{\text{Capture Key Recovery Attacks}} + \underbrace{\frac{d \cdot q_e + 8}{2^{256}}}_{\text{Capture Key Collision Attacks}} + \underbrace{\frac{1536 \cdot q_v}{2^{256}}}_{\text{Capture Key Recovery Attacks}} + \underbrace{\frac{(\sigma_e + q_e)^2}{2^{513}}}_{\text{Capture Block Collision Attacks}}$$

- The bound is tight $\rightarrow$ we give matching attacks for each term

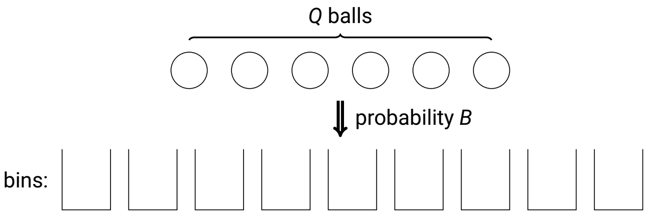
Multi-User Security Bound

$$\mathsf{Adv}^{\mathrm{muAE}}_{\mathrm{ChaCha20\text{-}Poly1305}[\pi]}(\mathcal{A}) \leq \frac{3 \cdot q_v(\ell_m + 1)}{2^{104}} + \frac{p \cdot (d + 512)}{2^{256}} + \frac{d \cdot q_e + 8}{2^{256}} + \frac{1536 \cdot q_v}{2^{256}} + \frac{(\sigma_e + q_e)^2}{2^{513}}$$

- The bound is tight $\rightarrow$ we give matching attacks for each term
- This bound can be used to tune the parameters of protocols using ChaCha20-Poly1305
- d : max number of times a same nonce is reused across different users during encryption ($\Rightarrow \mathcal{A}$ is called a d-repeating adversary)

$\rightarrow$ for non nonce-randomized schemes, $d = q_e$
$\rightarrow$ for nonce-randomized schemes, $d \ll q_e$ is bounded through a probabilistic balls-into-bins argument

Biased Balls-Into-Bins
Previous Results

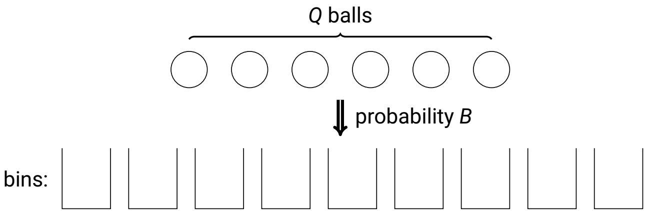


- Maximum load results for a slightly biased ball distribution B are given in [BHT18]

  $\rightarrow$ reused in [HTT18]:
  - To improve over the bounds in [BT16] for nonce-randomization
  - Extends security bounds of a classical AEAD scheme for d-repeating adversaries to its nonce-randomized version
  - Introduces a limiting term of 2^{-48} in the bound

Biased Balls-Into-Bins

Our Improvement



- We improve over [BHT18], by allowing any biased ball distribution B and number of balls Q, in addition to a tradeoff parameter between the maximum load and its probability

$\rightarrow$ We improve the prior bound for nonce-randomization:

- Replacing 2^{-48} with 2^{-192} in the bound for $d \leq 287$
- Improving the bound also for nonce-randomized AES-GCM
- In practice, more queries are allowed in protocols

Interpretation of the Bounds

Security Properties of ChaCha20-Poly1305

- The security profile of ChaCha20-Poly1305 is very different from AES-GCM:
  - Dominant term for AES256-GCM:

    $$\frac{\sigma \cdot B}{2^{128}} \rightarrow \text{corresponds to AES (the encryption component)}$$

  - Dominant term for ChaCha20-Poly1305:

    $$\frac{q_v \cdot \ell_m}{2^{104}} \rightarrow \text{corresponds to Poly1305 (the MAC component)}$$

  $\rightarrow$ protocols need to tune their parameter limits differently

- Rekeying does not improve the multi-user security of ChaCha20-Poly1305.

Summary

- We gave a new security analysis of ChaCha20-Poly1305:
  - in the Single-User setting: we gave a new proof of Procter's bound,
  - in the Multi-User setting: we gave a detailed analysis on par with that for AES-GCM.
- We described attacks to prove the tightness of every term in our multi-user security bound.
- We improved in the process the bound for nonce-randomized AES-GCM.
- We highlighted that the security limits of ChaCha20-Poly1305 are different from AES-GCM.
- We provide a simple way to strengthen the scheme by increasing the hash size.

Full version available soon on IACR ePrint

References I

[BHT18] Priyanka Bose, Viet Tung Hoang, and Stefano Tessaro. “Revisiting AES-GCM-SIV: Multi-user Security, Faster Key Derivation, and Better Bounds”. In: *EUROCRYPT 2018, Part I*. Ed. by Jesper Buus Nielsen and Vincent Rijmen. Vol. 10820. LNCS. Springer, Heidelberg, Apr. 2018, pp. 468–499. DOI: 10.1007/978-3-319-78381-9_18.

[BT16] Mihir Bellare and Björn Tackmann. “The Multi-user Security of Authenticated Encryption: AES-GCM in TLS 1.3”. In: *CRYPTO 2016, Part I*. Ed. by Matthew Robshaw and Jonathan Katz. Vol. 9814. LNCS. Springer, Heidelberg, Aug. 2016, pp. 247–276. DOI: 10.1007/978-3-662-53018-4_10.

[HTT18] Viet Tung Hoang, Stefano Tessaro, and Aishwarya Thiruvengadam. “The Multi-user Security of GCM, Revisited: Tight Bounds for Nonce Randomization”. In: *ACM CCS 2018*. Ed. by David Lie et al. ACM Press, Oct. 2018, pp. 1429–1440. DOI: 10.1145/3243734.3243816.

References II

[IOM12] Tetsu Iwata, Keisuke Ohashi, and Kazuhiko Minematsu. "Breaking and Repairing GCM Security Proofs". In: *CRYPTO 2012*. Ed. by Reihaneh Safavi-Naini and Ran Canetti. Vol. 7417. LNCS. Springer, Heidelberg, Aug. 2012, pp. 31–49. DOI: 10.1007/978-3-642-32009-5_3.

[Jea16] Jérémy Jean. *TikZ for Cryptographers*. https://www.iacr.org/authors/tikz/. 2016.

[LMP17] Atul Luykx, Bart Mennink, and Kenneth G. Paterson. "Analyzing Multi-key Security Degradation". In: *ASIACRYPT 2017, Part II*. Ed. by Tsuyoshi Takagi and Thomas Peyrin. Vol. 10625. LNCS. Springer, Heidelberg, Dec. 2017, pp. 575–605. DOI: 10.1007/978-3-319-70697-9_20.

References III

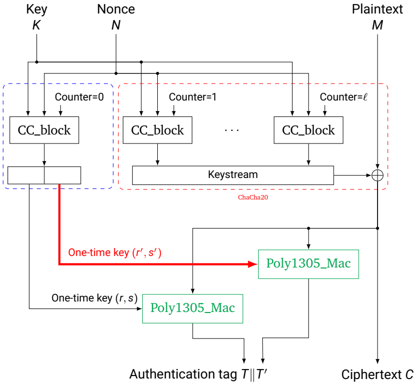
[MV04] David A. McGrew and John Viega. "The Security and Performance of the Galois/Counter Mode (GCM) of Operation". In: *INDOCRYPT 2004*. Ed. by Anne Canteaut and Kapalee Viswanathan. Vol. 3348. LNCS. Springer, Heidelberg, Dec. 2004, pp. 343–355.

[Niw+15] Yuichi Niwa et al. "GCM Security Bounds Reconsidered". In: *FSE 2015*. Ed. by Gregor Leander. Vol. 9054. LNCS. Springer, Heidelberg, Mar. 2015, pp. 385–407. DOI: 10.1007/978-3-662-48116-5_19.

[Pro14] Gordon Procter. *A Security Analysis of the Composition of ChaCha20 and Poly1305*. Cryptology ePrint Archive, Report 2014/613. https://eprint.iacr.org/2014/613. 2014.

Increasing the Hash Size

ChaCha20-cPoly1305



Proof Overview

H-Coefficient technique

For any **good transcript** τ it holds that:

$$\frac{\mathsf{P}_{\mathrm{real}}(\tau)}{\mathsf{P}_{\mathrm{ideal}}(\tau)} \geq 1 - \frac{2q_v}{2^t}.$$

For **bad transcripts**:

$$\Pr[\mathcal{T}_{\mathrm{ideal}} \in \mathsf{Bad}_1] \leq \frac{pd}{2^k}.$$

$$\Pr[\mathcal{T}_{\mathrm{ideal}} \in \mathsf{Bad}_2] \leq \frac{p \cdot 2\overline{(n-k)}^{\sigma_e}}{2^k} + \frac{1}{2^{n-k}}.$$

$$\Pr[\mathcal{T}_{\mathrm{ideal}} \in \mathsf{Bad}_3] \leq \frac{q_e(d-1)}{2^k}.$$

$$\Pr[\mathcal{T}_{\mathrm{ideal}} \in \mathsf{Bad}_4] \leq \frac{(\sigma_e + q_e)^2}{2^{n+1}}.$$

$$\Pr[\mathcal{T}_{\mathrm{ideal}} \in \mathsf{Bad}_5] \leq \frac{q_v \cdot 2\left(\overline{(n-k)}^p + \overline{2t}^p\right)}{2^k} + \frac{1}{2^{n-k-1}} + \frac{1}{2^{2t-1}}.$$

$$\Pr[\mathcal{T}_{\mathrm{ideal}} \in \mathsf{Bad}_6] \leq \frac{q_v}{2^t} + \frac{q_v \cdot c \cdot \ell_m}{2^t} + \frac{q_v \cdot 2 \cdot \overline{2t}^d}{2^k} + \frac{1}{2^{2t}}.$$