Private Message Franking with After Opening Privacy

Iraklis Leontiadis and Serge Vaudenay

LASEC, EPFL, Switzerland
{iraklis.leontiadis,serge.vaudenay}@epfl.ch

Abstract. Recently Grubbs *et al.* [GLR17] initiated the formal study of message franking protocols. This new type of service launched by Facebook, allows the receiver in a secure messaging application to verifiably report to a third party an abusive message some sender has sent. A novel cryptographic primitive: committing AEAD has been initiated, whose functionality apart from confidentiality and authenticity asks for a compact commitment over the message, which is delivered to the receiver as part of the ciphertext. A new construction CEP (Committing Encrypt and PRF) has then been proposed, which is multi-opening secure and reduces the computational costs for the sender and the receiver.
Despite the merits of the message franking protocols [GLR17], our observation which launched this work, is that all the designs be it compositional or the CEP construction, leak too much when the receiver needs to open the abusive message to the third party. Namely, the receiver opens the entire message along with the opening key to the third party, thus confidentiality of the message is entirely broken. Moreover, the opening of the entire message increases the communication cost of the protocol and in cases of big messages being exchanged (attachments, videos, multimedia files, etc.) it might be unnecessary. We provide to the best of our knowledge the first formal treatment of message franking protocols with minimum leakage whereby only the abusive blocks are opened, while the rest non-abusive blocks of the message remain private.
First we give a new definition for *multi-opening indistinguishability with partial opening* (MO-IND-PO), which forces an adversary to distinguish encryptions of abusive blocks. We then design and analyze two protocols CEP-AOP1 (Committing Encrypt and PRF with After Opening Privacy) and CEP-AOP2, which adhere to the new privacy definition. As a side contribution we show a multi-opening secure CEP2 construction using only one PRF evaluation over the message, in a weaker but meaningful security model, relying only on standard assumptions of the underlying symmetric primitives.

Keywords: message franking protocols, abusive reports, messaging applications, partial opening, privacy, secure communication

1 Introduction

We are witnessing the transition to a digital messaging society. Billions of users are using messaging application to communicate with other end users. The majority choose messaging applications over the Internet with no extra charging policy like Facebook messaging [Fac], Whatsapp [Wha], Signal [Sig], Telegram [Tel], Viber [Vib], etc. The security goals of messaging applications is end to end confidentiality and integrity: no intermediate party by observing exchanged transcripts over public or private channels can compromise integrity or confidentiality. However, it seems that these are not the only required security guarantees for secure messaging. This model however fails to capture adversarial senders, who send illegal harassing content [Cal18]. Recently, Facebook introduced the notion of *message franking*, which guarantees that when a sender sends a harassing message to a receiver, the latter can verifiably report it to Facebook.

Facebook messaging protocol for message franking allows a receiver to verifiability open an abusive message to Facebook, without being able to report fake messages. At a high level the protocol lies on an authenticated encryption scheme AE to provide confidentiality and authenticity of the messages and on a pseudorandom function (PRF), in order the sender to commit to the sent message M. The PRF should enjoy the property of collision resistance in order to avoid malicious openings of the sender to fake messages. Grubbs *et al.* [GLR17] were the first to formalize the security definition for message franking and showed 3 compositional designs following the Encode-then-Encipher [BR00], Encrypt-then-Mac [BR00], Mac-then-Encrypt compositions, which are only single opening secure, meaning that after the opening the confidentiality-integrity of the messages is not preserved and the two users should share new keys. Those protocols need 5 passes over the message for encryption and decryption and 2 for verification.

The authors then presented a multi-opening secure scheme dubbed CEP for Committing Encrypt-and-PRF, which uses a nonce-based pseudorandom generator (PRG) and two PRF's. The scheme needs 3 passes for encryption and decryption and 2 for verification. Recently, Dodis *et al.* [DGRW18] revealed a flaw on the message franking protocol of Facebook for attachments. The authors also proposed a new neat construction based on a new primitive: *encryptment*, which is simply a committing AEAD, but only single pass secure and is based on hash function chaining (HFC). The final committing AEAD is based on the proposed encryptment primitive but confidentiality is based on non-standard assumption of the related-key-attack-secure (RKA) property of the underlying PRF. Despite the valuable merits of those works, all those designs suffer from **1)** intensive privacy leakage to the router and **2)** augmented, unnecessary communication overload. The deficiencies can be summarized as follows:

1. A sender sending a message M consisting of abusive information is opened at its entire form to the router. However, it might be the case the message itself holds private information the receiver does not want to reveal to the router.

2. In cases with large messages such as attachments videos and multimedia files it might be unnecessary to open all the blocks to the router. The abusive content is spotted in some block of the entire big message and thus opening all the blocks of the message to the router skyrockets the communication complexity of the protocol.

The above realistic scenarios are not captured by current state of the art [GLR17]. The problem arises from the treatment of messages as singleton objects during the protocol execution. The entire message is given as input to the encryption algorithm and the same message feeds the committing primitive–the PRF. As we are in the symmetric setting, the internals of the encryption algorithm and the committing primitive treat the messages as a set of blocks. As such during the opening procedure the receiver of a possible abusive message is obliged to open the entire message–all the blocks. There is little freedom left to the receiver at this point as private and abusive blocks will all be revealed to the router. For example for an m-block message M: $|b_1|b_2|\ b_3\ |\ b_4\ |\ b_5\ |\ b_6\ |\ b_7\ |\ b_8\ |\ b_9\ |\ b_{10}\ |\ \cdots\ |\ b_m|$, let the green blocks $(b_5 \dots b_{10})$ consist the abusive information and the red ones the private ones $(b_1 \dots b_4, b_{11} \dots b_m)$. Current message franking designs give up privacy entirely for all the blocks. In this paper we seek to answer the following question:

"*Can we improve the privacy and subsequently the communication complexity of a message franking protocol, after the open-report procedure to the router by opening only the necessary blocks?*"

We answer this question with an affirmative response. First we provide a new formal model to capture the privacy of message franking protocols and then we instantiate two message franking protocols with after opening privacy.

Contributions More specifically the contributions of this work are as follows:

- We introduce a more realistic privacy definition for abusive message report enhancing previous definitions, called After Opening Privacy AOP. Intuitively, AOP guarantees that if a message with $|M|/n$ blocks, consists of some α abusive blocks and some β non-abusive ones, where $\alpha + \beta = |M|/n$, then after the opening procedure the confidentiality of the β private blocks is preserved. We end up designing to the best of our knowledge the first private message franking protocols: CEP-AOP1 and CEP-AOP2, which achieve the novel notion of after opening security AOP (cf. Table 1).
- As a second contribution, we revisit the security definition for multi-opening ciphertext integrity as presented in [GLR17] and we realize a CEP scheme with only 2 passes for encryption, decryption and verification, relying only on standard assumption of the underlying symmetric primitives. Namely the authors in [GLR17] showed that using one key to commit to the message allows for forgery attacks. This is doable because an adversary, according to the game, can obtain a ciphertext on her own without querying the Enc oracle. After that, it receives the opening key from the Dec oracle and finally forges a valid commitment during the challenge phase. To circumvent

the attack the authors proposed to commit to the ciphertext with a PRF using a key which is kept secret between the sender and the receiver. After careful altering of the game, which results in a weaker but meaningful security model for integrity, we present the CEP2 scheme without the need of an extra PRF and key. We change the game to capture calls to the Dec oracle through some bookkeeping, thus allowing the challenger during the challenge phase to realize when the adversary called the Dec oracle without calling the Enc oracle and discarding the attack as non-valid. In contrast with the [DGRW18] scheme, which is also 2 passes in the same strong security model for integrity as [GLR17], our solution lies only on standard assumptions and not on related-key-attacks security of the PRF.

Outline The rest of the paper is organized as follows: In Section 2 we depict the message franking protocol of facebook and the formal security definitions as formalized by Grubbs *et al.* [GLR17]. We also describe the commit then authentication (CEP) scheme. We continue in Section 3 with a new security definition (MO-nCTXT2) for multi-opening ciphertext integrity for the existing CEP scheme, which allows to tweak CEP in a way with no need for a second PRF, thus a more efficient construction with less passes over the initial message and the ciphertext. We also describe our new optimal design CEP2 and its security analysis in a provable way. In Section 4 we present the definitional framework for our novel primitive: Committing Nonce based Authenticated Encryption with Partial Opening (CEPO) in order to instantiate our two message franking protocols with after opening privacy. Furthermore, we describe the new privacy definition for message franking with *after opening privacy* (AOP) with a game played between an adversary and a challenger. We present our two protocols CEP-AOP1 and CEP-AOP2 which adhere to AOP in Sections 5 and 6 with their provable secure analysis. Before concluding and presenting future work directions in Section 8 we depict related work in Section 7.

2 Facebook Franking

Facebook franking protocol (cf. Figure 1) operates as follows. Users run a key-agreement protocol for a common secret encryption key k. The key k is agreed following the SIGNAL protocol specificiations [Fac16]. It is out of the scope of the current manuscript to communicate the details of the key exchange protocol, but we assume a secure key exchange running between the sender $\mathcal{S}$ and the receiver $\mathcal{R}$ in order to agree upon the symmetric key k. The sender $\mathcal{S}$ runs a key generation algorithm to generate an HMAC key sk_f and evaluates the HMAC on the concatenation $M\|\mathsf{sk}_f$ to compute the image C_2. It then encrypts $M\|\mathsf{sk}_f$ using an authenticated encryption algorithm Enc, which takes as input header data H as well and results in C_1. $\mathcal{S}$ forwards C_1, C_2 to Facebook, who in turn evaluates HMAC, keyed by fk on $C_2\|md$, where $md \leftarrow \mathcal{S}\|\mathcal{R}\|\text{time}$ and sends $a \leftarrow \mathsf{HMAC}_{\mathsf{fk}}(C_2\|md)$, C_1, C_2 to the receiver $\mathcal{R}$. $\mathcal{R}$ uses its symmetric key k to decrypt C_1 in M, sk_f and verifies the correctness of C_2 using sk_f and the HMAC. If everything is correct it accepts the message M as valid. Later on, $\mathcal{R}$ decides

Scheme	AOP	MO	SB	RB	Enc	Dec	Verify
EtE[GLR17]	✗	✗	✓	✓	-	-	-
EtM[GLR17]	✗	✗	✓	✓	$2+1$	$2+1$	$2+1$
MtE[GLR17]	✗	✗	✓	✓	$2+1$	$2+1$	$2+1$
CtE1[GLR17]	✗	✓	✓	✓	$3+1$	$3+1$	$1+1$
CtE2[GLR17]	✗	✓	✓	✓	$3+2$	$3+2$	$1+1$
CEP[GLR17]	✗	✓	✓	✓	$2+1$	$2+1$	$1+1$
HFC[DGRW18]	✗	✓	✓	✓	2	2	2
CEP2	✗	✓	✓	✓	2	2	2
CEP-AOP1	✓	✓	✓	✓	$1+m$	$1+m$	α
CEP-AOP2	✓	✓	✓	✓	$1+m$	$1+m$	$\log m$

Table 1: Comparison for EtE, EtM, MtE, CtE1, CtE2, CEP, HFC, CEP2, CEP-AOP1 and CEP-AOP2. AOP is for after opening privacy, MO stands for multi-opening security, SB for sender binding and RB for receiver binding. The concrete numbers under the protocol algorithms demonstrate the number of passes over the message. m denotes the number of blocks for a message M and $\alpha \leq m$ is the number of abusive blocks in M.

to flag the message M as abusive. $\mathcal{R}$ to convince Facebook that the message M sent by $\mathcal{S}$ is abusive sends to Facebook $(M, \mathsf{sk}_f, md, a)$. Facebook computes $a' \leftarrow \mathsf{HMAC}_{\mathsf{fk}}(\mathsf{HMAC}_{\mathsf{sk}_f}(M \| \mathsf{sk}_f) \| md)$ and verification is correct if a' matches with a.

2.1 Committing Nonce based Authenticated Encryption

Regardless its simplicity, Facebook abusive message reporting protocol incurs high computation complexity. Assuming Enc is an authenticated encryption scheme then for encryption the sender has to process the message M, 5 times, the receiver passes the message 5 times in order to decrypt and the router-Facebook verifies whether the message has been truly sent by the sender $\mathcal{S}$ to the receiver $\mathcal{R}$ in 2 passes. Grubbs *et al.* [GLR17] proposed a new protocol dubbed CEP, which reduces the number of passes. To instantiate their protocol they put forth the definition of committing authenticated encryption (CE).

A CE scheme consists of four algorithms $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Verify})$, associated with a message space $\mathcal{M} \in \Sigma^*$, a key space $\mathcal{K} \in \Sigma^*$, a nonce space $\mathcal{N} \in \Sigma^*$, a header space $\mathcal{H} \in \Sigma^*$, a ciphertext space $\mathcal{C} \in \Sigma^*$, an opening space $\mathcal{O} \in \Sigma^*$ and a franking space $\mathcal{T} \in \Sigma^*$. The four algorithms are defined as follows:

- $\mathsf{k} \leftarrow\!\!\$\ \mathsf{KGen}(1^\lambda)$: A randomized algorithm, which outputs a secret key $\mathsf{k} \in \mathcal{K}$, on input a security parameter λ in its unary form.
- $(C_1, C_2) \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{k}, H, N, M)$: The encryption algorithm, which is deterministic, takes as input a key, a header, a nonce and a message $(\mathsf{k}, H, N, M) \in (\Sigma^*)^4$ and outputs $(C_1, C_2) \in \mathcal{C} \times \mathcal{T}$ or $\perp$. C_1 will be usually referred to as the ciphertext and C_2 as the commitment or tag.

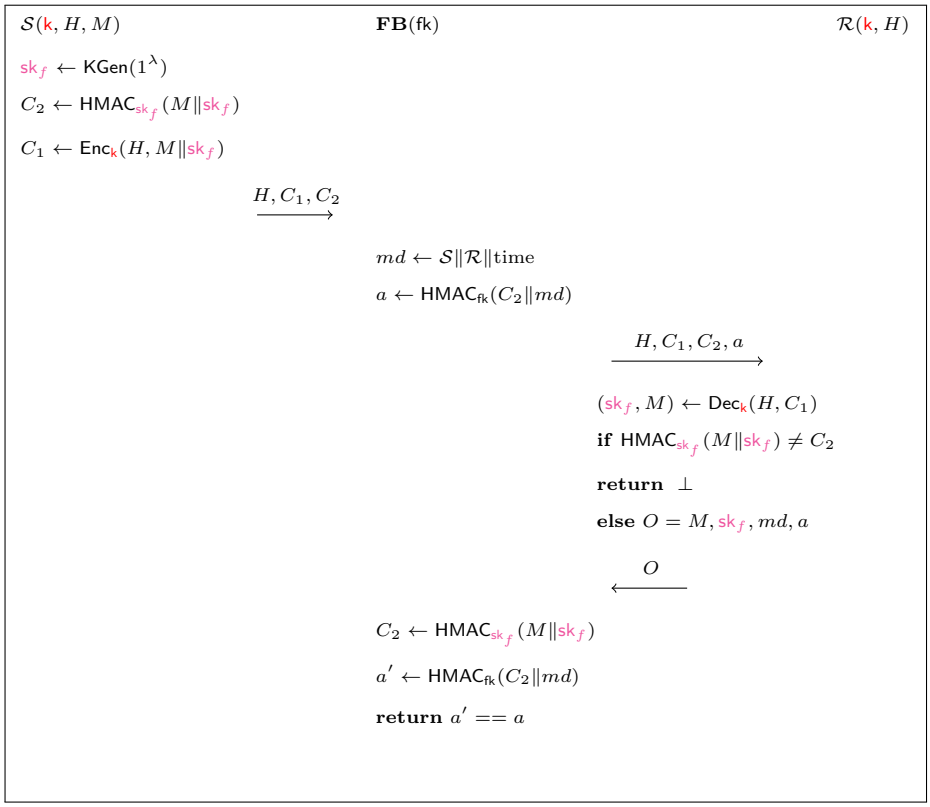

Fig. 1: Facebook message franking protocol

- $(\mathsf{sk}_f, M) \leftarrow \mathsf{Dec}(\mathsf{k}, H, N, C_1, C_2)$: The decryption algorithm Dec is a deterministic algorithm, which takes as input $(\mathsf{k}, H, N, C_1, C_2) \in (\Sigma^*)^5$ and outputs a message $M \in \mathcal{M}$ with an opening key $\mathsf{sk}_f \in O$, or $\perp$.
- $0, 1 \leftarrow \mathsf{Verify}(H, M, \mathsf{sk}_f, C_2)$. This deterministic algorithm takes as input $(H, M, \mathsf{sk}_f, C_2) \in (\mathcal{H} \times \mathcal{M} \times \mathcal{O} \times \mathcal{T})$ and outputs 1 if verification is successful and 0 otherwise.

For encryption and decryption algorithms we sometimes omit the first input: the secret key k, and we will write $\mathsf{Enc}_{\mathsf{k}}(\cdot,\cdot,\cdot)$ and $\mathsf{Dec}_{\mathsf{k}}(\cdot,\cdot,\cdot,\cdot)$ or $\mathsf{Enc}(\cdot,\cdot,\cdot)$ and $\mathsf{Dec}(\cdot,\cdot,\cdot,\cdot)$. We also write $\mathsf{Enc}_{\mathsf{k}}(\cdot,\cdot,\cdot)[1]$ and $\mathsf{Enc}_{\mathsf{k}}(\cdot,\cdot,\cdot)[2]$ to denote C_1 and C_2 and accordingly $\mathsf{Dec}_{\mathsf{k}}(\cdot,\cdot,\cdot,\cdot)[1], \mathsf{Dec}_{\mathsf{k}}(\cdot,\cdot,\cdot,\cdot)[2]$ for sk_f and M.

A CE is correct if it adheres to 1)*decryption correctness* as in correctness for encryption schemes: $\forall(\mathsf{k}, H, N, M) \in (\mathcal{K} \times \mathcal{H} \times \mathcal{N} \times \mathcal{M})$ it is true that:

$$\underbrace{\Pr[\mathsf{Dec}(\mathsf{k}, H, N, \underbrace{\mathsf{Enc}_{\mathsf{k}}(H, N, M)}_{C_1, C_2})[2] = M]}_{M} = 1$$

and 2) commitment correctness: if $\forall(\mathsf{k}, H, M) \in (\mathcal{K} \times \mathcal{H} \times \mathcal{M})$ it is true that:

$$\Pr[\mathsf{Verify}(H, M, \underbrace{\mathsf{Dec}_{\mathsf{k}}(H, N, C_1, C_2)[1]}_{\mathsf{sk}_f}, \underbrace{\mathsf{Enc}_{\mathsf{k}}(H, N, M)[2]}_{C_2}) = 1] = 1$$

The CEP protocol [GLR17] (c.f. Figure 2) operates as follows: The sender shares a secret key k with the receiver. Afterwards $\mathcal{S}$ selects uniformly at random a keystream P of size $m+2n$ using a pseudorandom generator G which takes as input a key k, a nonce N and a desired output length l. G can be instantiated with a block cipher E in counter mode with $IV \leftarrow E_{\mathsf{k}}(N)$: the $i^{th}n$ bit block is the output of $E_{\mathsf{k}}(IV+i)$ for messages in $GF(2^n)$. The $2n$ bits of the keystream are used as keys for two PRF's $\mathsf{F}^{cr}_{P_0}$ and F_{P_1}, where F_{P_0} should be collision resistant. The rest m bits of P are used as a xor based one-time-pad to encrypt the message M. The reason for the second PRF is to achieve multi-opening security. However, as we will show later, we can capture trivial attacks with some extra bookkeeping at the security definition of ciphertext integrity without degrading the security model and without the need for a second PRF. For decryption the receiver $\mathcal{R}$ runs the deterministic G on input the secret shared key k to derive the same keystream P. It decrypts C_1 and then checks the integrity of both C_2 and T. Whenever $\mathcal{R}$ receives an abusive message M, it sends to the router $M, \mathsf{sk}_f = P_0$ and the router checks whether $C_2 == \mathsf{F}^{cr}_{\mathsf{sk}_f}(H\|M)$. If that is the case the protocol continues with the second step of verification from Figure 1, which incorporates the validity of a, where a the tag computed on $C_2\|\mathcal{S}\|\mathcal{R}\|timestamp$ keyed with the secret key of the router. Overall the CEP results in 3 passes for encryption and decryption and 2 pases for the verification.

$\mathsf{CEP\text{-}Enc}_{\mathsf{k}}(H, N, M)$:

$m \leftarrow |M|/n$

$P \leftarrow G(\mathsf{k}, N, 2n + nm)$

$C \leftarrow (P_2\| \cdots \|P_{m+1} \oplus M)$

$C_2 \leftarrow \mathsf{F}^{cr}_{P_0}(H\|M)$

$T \leftarrow \mathsf{F}_{P_1}(C_2)$

return $(C_1 = C\|T, C_2)$

$\mathsf{CEP\text{-}Dec}_{\mathsf{k}}(H, N, C_1, C_2)$:

parse $C_1 = C\|T$

$m \leftarrow |C|/n$

$P \leftarrow G(\mathsf{k}, N, 2n + nm)$

$M \leftarrow (P_2\| \cdots \|P_{m+1} \oplus C)$

$C'_2 \leftarrow \mathsf{F}^{cr}_{P_0}(H\|M)$

$T' \leftarrow \mathsf{F}_{P_1}(C'_2)$

if $T \neq T' \vee C'_2 \neq C_2$ **then**

return $\perp$

return (P_0, M)

$\mathsf{CEP\text{-}Verify}(H, M, \mathsf{sk}_f, C_2)$:

$C'_2 \leftarrow \mathsf{F}^{cr}_{\mathsf{sk}_f}(H\|M)$

if $C'_2 \neq C_2$ **then**

return $\perp$

return 1

Fig. 2: CEP algorithms

2.2 Message Franking Protocol Security Definitions

In terms of security message franking protocols as modeled by Grubbs *et al.* [GLR17] are formalized within four flavors of security. The folklore confidentiality notion, which protects the privacy of the exchanged messages between $\mathcal{S}$ and $\mathcal{R}$, the integrity protection of the message itself against active adversaries and two novel definitions for binding.

Binding Security The first one: *Sender binding* (s-BIND) *locks* $\mathcal{S}$ from sending an abusive message M' which cannot be verifiably reported by $\mathcal{R}$ to the router and the second one: Receiver Binding r-BIND protects $\mathcal{S}$ from a malicious $\mathcal{R}$ who tries to convince the router about a claimed abusive message M', which has not been truly sent by $\mathcal{S}$. r-BIND protects the sender from faulty being accused as a sender of an abusive message which has never been sent. It is relevant with the symmetric setting as simple encryption with a symmetric cipher does not protect the sender, e.g: as sender and receiver act honestly during the authenticated key exchange phase they both share the symmetric encryption key k. A malicious $\mathcal{R}$ can encrypt any message she wants and claimed that it has been sent by $\mathcal{S}$. As such, to achieve receiver binding a careful treatment of cryptographic primitives other than integrity tags and ciphertexts is needed.

s-BIND definition is depicted in Figure 3 with the notion of a game. The game starts with the adversary $\mathcal{A}$ selecting a key k, a header H, C_1 and C_2. At the next step the challenger (benign entity) of the game tries to decrypt C_1, C_2 with the Dec algorithm and gets sk_f and M'. If the decrypted message is empty the game halts. Otherwise, the challenger runs the Verify algorithm on input $H, M', \mathsf{sk}_f, C_2$. Finally, $\mathcal{A}$ wins the game $\iff$ the output of the verification algorithm is 0, which means that the sender managed to send a message M', which cannot be verified by the router as abusive. The advantage of an adversary $\mathcal{A}$ against sender binding equals the probability of the game to s-BIND to output 1:

$$\mathsf{Adv}^{\mathrm{s-bind}}_{\mathsf{CEP}}(\mathcal{A}) = \Pr[\text{s-BIND} \Rightarrow 1]$$

For the Receiver Binding Game (r-BIND) (cf. Figure 3) $\mathcal{A}$ first outputs $(H, M, \mathsf{sk}_f), (H', M', \mathsf{sk}_f'), C_2$. The challenger runs twice the Verify algorithm on input $(H, M, \mathsf{sk}_f, C_2)$ and $(H', M', \mathsf{sk}_f', C_2)$ and outputs b and b'. If $(H, M) == (H', M')$ then the game outputs 0 and halts as that models the fact that the adversary $\mathcal{A}$ impersonating the receiver $\mathcal{R}$ did not manage to find a different header and message that will be successfully verified by the router. Otherwise if $(H, M) \neq (H', M')$ and $b = b' = 1$ meaning the router verified them successfully then the game returns successfully the value 1. Similarly with the sender binding notion, the advantage of an adversary $\mathcal{A}$ against receiver binding equals the probability of the game s-BIND to terminate without outputting 1:

$$\mathsf{Adv}^{\mathrm{r-bind}}_{\mathsf{CEP}}(\mathcal{A}) = \Pr[\text{r-BIND} \Rightarrow 1]$$

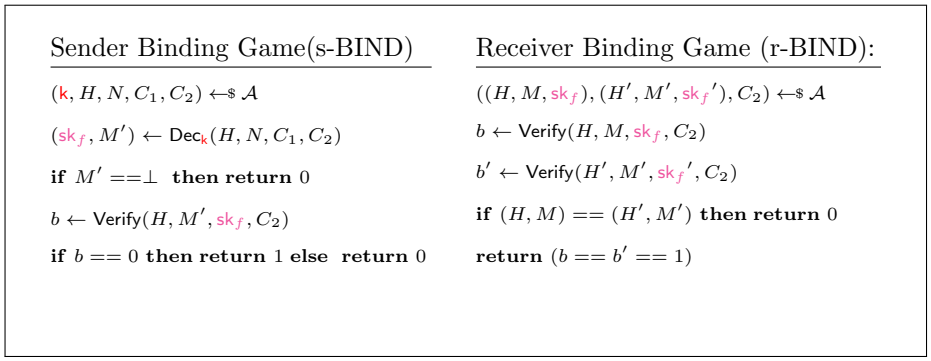

Fig. 3: Sender and Receiver binding games

Confidentiality and Integrity Confidentiality is modeled with a Real-or-Random game whereby the adversary tries to distinguish with non-negligible probability whether it receives ciphertexts from the real message franking encryption algorithm or it receives encryption of uniformly random bitstrings of the same length as the real ones. As the construction uses nonce-based authenticated encryption with authenticated data the authors [GLR17] lifted the corresponding confidentiality definition tailored for nonce-based encryption with authenticated header to the message franking setting (cf. Figure 4). In all games $\mathcal{A}$ is nonce-respecting to avoid trivial wins: E.g: $\mathcal{A}$ asks the encryption of a message M with nonce N from the Enc oracle and gets as a response the corresponding ciphertext C_1, C_2. Afterwards, it calls the challenge oracle on input (H, N, M) and it gets back either the faithful encryption of M or the encryption of same length random bit string. To win the game it compares the result with C_1, C_2 from the encryption oracle.

We slightly change the game as first defined by Grubbs *et al.* [GLR17] in order to be more precise and avoid ambiguities during its interpretation: It is not clear why the decryption oracle should only allow decryptions made from encryption oracle. By doing so, the game correctly forbids decryption for the challenge ciphertext but it also weakens the model because it forces adversary to make a query to the encryption oracle to learn a ciphertext, but the adversary $\mathcal{A}$ may compute correctly a ciphertext at its own. As such we keep track of the challenge in a variable y_1 and check at the decryption oracle if the input equals y_1. In that way we give more freedom to $\mathcal{A}$ to come up with a valid ciphertext. Second we add a variable y_0 in order to model the nonce respecting adversary. Finally we control the number of the challenges to equal 1.

The advantage of an adversary $\mathcal{A}$ against the confidentiality of a message franking protocol is:

$$\mathsf{Adv}^{\mathrm{mo-nror}}_{\mathsf{CEP}}(\mathcal{A}) = |\Pr[\text{MO-nREAL} \Rightarrow 1] - \Pr[\text{MO-nRAND} \Rightarrow 1]|$$

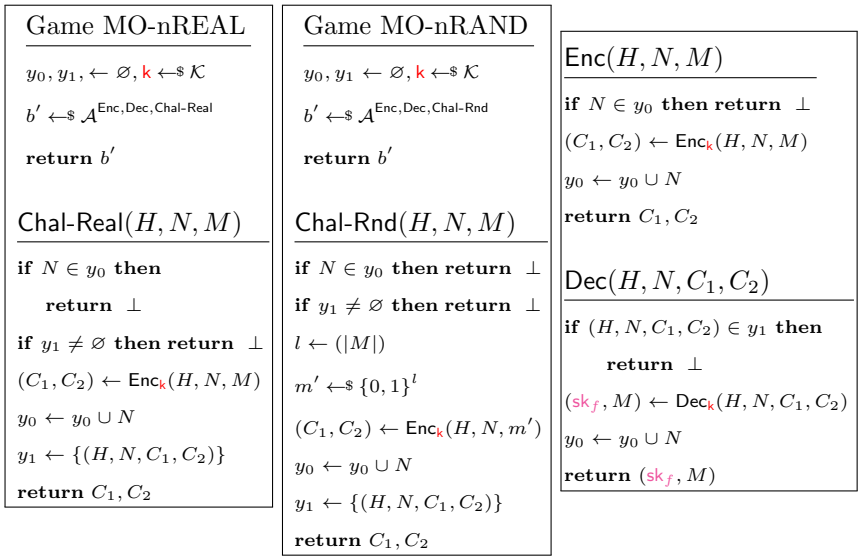

Fig. 4: MO-nREAL and MO-nRAND games

The integrity mechanism protects the integrity of the exchanged messages either from external adversaries, who try to manipulate the messages, or from internal adversaries trying to forge a message in order to convince the router for a delivery of message. Figure 5 demonstrates the game played between an adversary $\mathcal{A}$ and a Challenger. $\mathcal{A}$ has access to the Enc, Dec oracles, which encrypt and decrypt arbitrary messages using an underlying nonce-based symmetric encryption algorithm. The adversary wins the game if it manages or present a ciphertext in the Challenge oracle, which can be successfully decrypted and has not been given as input to the Dec previously. The success probability of $\mathcal{A}$ winning the integrity game MO-nCTXT equals:

$$\mathsf{Adv}^{\mathrm{mo-nctxt}}_{\mathsf{CEP}}(\mathcal{A}) = \Pr[\text{MO-nCTXT} \Rightarrow 1]$$

3 Multi-Opening Security with one PRF

As a first observation we show how to achieve multi-opening security with respect to ciphertext integrity without the need of the second PRF evaluation over the PRF evaluated on the message M in a weaker security model but realistic in the case of message franking. This tweak results in an efficient CEP with only 2 of passes for encryption and decryption and 1 for verification. According to Grubbs *et al.* [GLR17], during Enc if the protocol ends by evaluating the PRF on the message: $C_2 \leftarrow \mathsf{F}^{cr}_{P_0}(H\|M)$ (i.e. the tag T is dropped) and during Dec the receiver only checks whether $C_2 = \mathsf{F}^{cr}_{P_0}(H\|M')$, where $M' \leftarrow (P_2\| \ldots \|P_{m+1}) \oplus C_1$ then there is an attack, which breaks the MO-nCTXT security guarantee. The attack

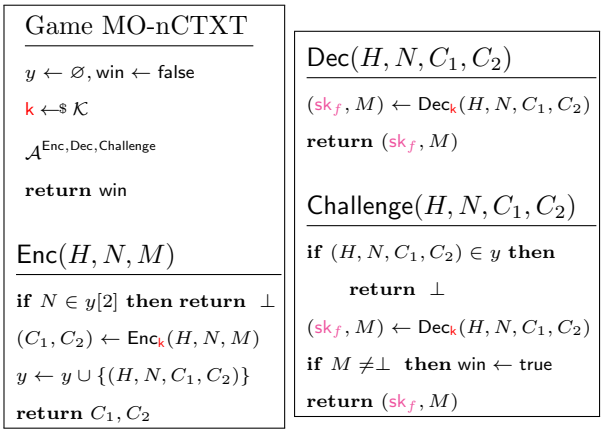

Fig. 5: Game MO-nCTXT and oracles

is described informally [GLR17, Section 7, p15] with some ambiguity and here we give the two interpretations which result in the same attack:

1. The adversary obtains a valid ciphertext (C_1, C_2) in the protocol execution (passive observer), thus it does not call the $\mathsf{Enc}(H, N, M)$ oracle and consequently the list y[1] for bookkeeping does not contain the tuple (H, N, C_1, C_2). Afterwards it calls the Dec oracle on input (H, N, C_1, C_2) to obtain the opening key $\mathsf{sk}_f = P_0$. With sk_f it can compute $\mathsf{F}^{cr}_{P_0}(H\|M')$ on arbitrary message M', which can be accepted by the $\mathsf{Challenge}$ oracle, because the tuple (H, N, C_1, C_2) is not in the y list.
2. A second interpretation of the attack contains an adversary $\mathcal{A}$, who calls the Enc oracle on input (H, N, M) and receives C_1, C_2. The oracle also stores the tuple (H, N, C_1, C_2) in the y list. $\mathcal{A}$ then calls the Dec oracle on input (H, N, C_1, C_2) to obtain the key sk_f as with the aforementioned first case. Afterwards, for $\mathsf{sk}_f = P_0$ it computes $C_2' \leftarrow \mathsf{F}^{cr}_{P_0}(H\|M')$ and presents $(H, N, C_1 \oplus M \oplus M', C_2')$ to the $\mathsf{Challenge}$ oracle, which sets $\mathsf{win} \leftarrow \mathsf{true}$ because (H, N, C_1, C_2') is not in the y list and decrypts well to M' with sk_f.

In either cases the problem arises from the fact that there is no bookkeeping at the Dec oracle. Thus, any time $\mathcal{A}$ obtains a valid ciphertext for any message M, it can ask for the opening key sk_f and evaluate the $\mathsf{PRF} : \mathsf{F}^{cr}_{\mathsf{sk}_f}$ on input any message at its choice. Then, at the $\mathsf{Challenge}$ oracle check, the tuple H, N, C_1, C_2' will never be in the list y, because C_2' has been computed locally by $\mathcal{A}$. The authors in [GLR17] proposed as a solution the use of a second $\mathsf{PRF} : \mathsf{F}$, which is used as follows: During encryption after obtaining the tag $C_2 \leftarrow \mathsf{F}^{cr}_{P_0}(H\|M)$ the sender uses a second key P_1 to evaluate another $\mathsf{PRF} : T \leftarrow \mathsf{F}_{P_1}(C_2)$, which takes as input C_2. During decryption the receiver accepts the ciphertext when

[1] The y list corresponds to Figure 4, from [GLR17]

both C_2 and T are correct and during opening it gives only the P_0 as a key to the router.

Here we show how to achieve multi-opening ciphertext integrity in a strictly weaker but meaningful model without the need for the second PRF and key, which rends the protocol more efficient and despite being weaker it is more realistic. Namely, in the MO-nCTXT game the adversary can compromise opening key P_0, which is at the same time the integrity key for the PRF. However, in a typical ciphertext integrity game the adversary is not compromising tagging integrity secret key when she is asked to forge a ciphertext. Thus, when the adversary is calling the decryption oracle to get the plaintext and the opening key $\mathsf{sk}_f = P_0 =$ the PRF key, we can *silence* the oracle with an abort in case the Challenge oracle is called with input a nonce N part of a decryption query. This abort makes the new security game MO-nCTXT2 weaker because of this restriction, but still meaningful in a real world model of a messaging application, where the receiver is reporting with the opening key and there is no delegation of the procedure to a third party.

First we change the security game for MO-nCTXT2 (cf. Figure 6). Namely, the new game keeps track of N tuples in a new list l and Challenge sets win $\leftarrow$ true when the decrypted message is not empty and N is not in l. Notice that the inclusion in the tuple of information related with the ciphertext C_1 and the header H does not give any advantage to the game to detect an adversary who has called the Dec oracle on input (H, N, C_1, C_2). $\mathcal{A}$ can call Dec to learn (sk_f, M), then it deduces $C_1 \oplus M = P_0 \dots P_m$, where $P_0 \dots P_m$ is the keystream. $\mathcal{A}$ then picks H', M' of the same length as H, M and queries the Challenge oracle on input $H', N, C_1 = (P_1 \dots P_m) \oplus M', C_2' = \mathsf{F}^{cr}_{P_0}(H' \| M')$. Thus, an adversary by learning any valid tuple of $M, \mathsf{sk}_f, C_1, C_2$ it can manipulate C_1 and C_2 to meaningful C_1' and C_2' for another message M' bypassing any check in the l list. The sufficient condition to capture trivial attacks is to forbid calls on the Challenge oracle on input a specific nonce N, which was part of a decryption query through the Dec oracle. Moreover, to avoid trivial wins whereby $\mathcal{A}$ passively listens the communication channel and submits H, N, C_1, C_2 to the Challenge oracle, we assume that any valid H, N, C_1, C_2 tuple has been produced by a call to the Enc oracle which keeps track of those tuples in y list.

With those tweaks the attack which has been described before regarding MO-nCTXT will be discarded with the membership checking at the list. We present in Figure 7 the new CEP2 scheme which is MO-nCTXT2 secure without the need of a second PRF. The changes with respect to the old CEP scheme are the deletion of 1) the second PRF evaluation during encryption, 2) the second key P_1 and 3) the second PRF check during decryption.

Obviously the security for MO-nCTXT2 implies MO-nCTXT : MO-nCTXT $\Rightarrow$ MO-nCTXT2 or to put it differently from an adversary breaking MO-nCTXT2 with a forgery we can implement a reduction to another adversary against MO-nCTXT by simply outputting the same forgery: $\text{MO-nCTXT2}_{\mathsf{breaking}} \Rightarrow \text{MO-nCTXT}_{\mathsf{breaking}}$. The other direction is not doable because a forgery for the MO-nCTXT security definition may not give a valid

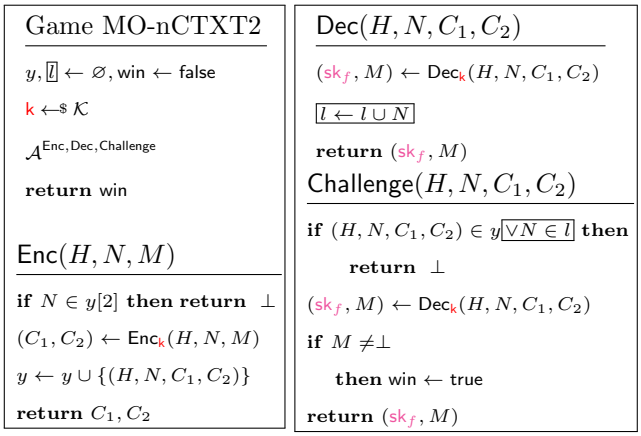

Fig. 6: Game MO-nCTXT2 and oracles. Boxed elements denote the differences with the MO-nCTXT game.

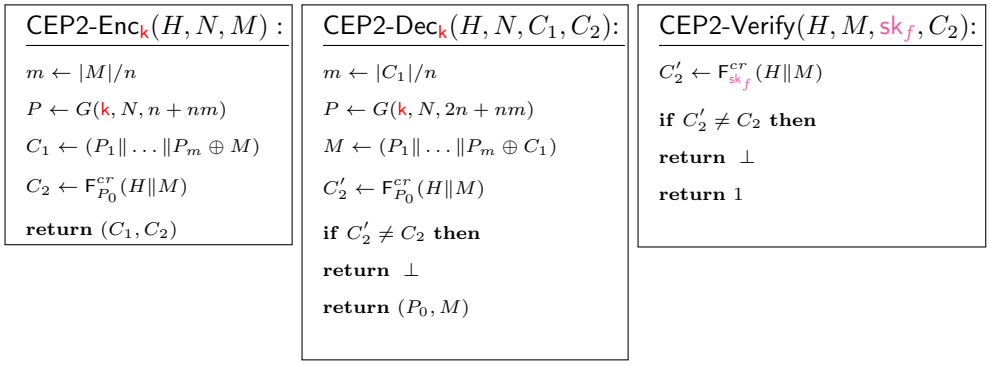

Fig. 7: CEP2 algorithms

forgery for the MO-nCTXT2 definition due to the restrictions of the game regarding adversarial behavior.

As we change the original CEP scheme by omitting the second PRF we prove the confidentiality, integrity and binding notions for CEP2. Intuitively the proofs follow the proofs for the CEP protocol omitting game transitions due to the second PRF : F since it is not used in CEP2. Interestingly we are getting tighter bounds for confidentiality and the same for integrity, and binding assuming F^{cr} is a also a secure MAC.

Theorem 1 (CEP2 Ciphertext integrity). *Let* CEP2 *be a message franking protocol instantiated with a pseudorandom generator G, a pseudorandom function F^{cr}. Let $\mathcal{A}$ be a* MO-nCTXT2 *adversary issuing $q = q_{\mathsf{Enc}}, q_{\mathsf{Dec}}, q_{\mathsf{Challenge}}$ queries with time complexity t. Then there exist adversaries $\mathcal{B}$ and $\mathcal{C}$ making each $q_{\mathsf{PRG}} = q_{\mathsf{Enc}} + q_{\mathsf{Dec}} + q_{\mathsf{Challenge}}$ and $q_{\mathsf{PRF}} = q_{\mathsf{Enc}} + q_{\mathsf{Dec}} + q_{\mathsf{Challenge}}$ queries in*

time complexity t such that:

$$\mathsf{Adv}^{\text{mo-nctxt2}}_{\mathsf{CEP2}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{prg}}_{G}(\mathcal{B}) + \sum_{j=1}^{q} \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{F}^{cr}}(\mathcal{C}_j)$$

Proof. Proof follows the game hoping technique through a series of game transitions, detectable with negligible probability according to the assumptions of a secure PRNG G and a $\mathsf{PRF} : \mathsf{F}^{cr}$. The proof inherits similarities with the CEP ciphertext integrity as instead of having $F^{cr} \circ F$ as a secure MAC we have $MAC = F^{cr}$. We assume without loss of generality that all queries (H, N, C_1, C_2) to the Dec oracle are in the y list or that $N \in l$. Otherwise we can use the $\mathsf{Challenge}$ oracle. Moreover, we assume that the game stops as soon as $\mathsf{win} = \mathsf{true}$. Hence, $\mathsf{Challenge}$ always returns $\perp$.

Let G_0 be the original game MO-nCTXT2 and G_1 equivalent with G_0 with one difference: calls to G are replaced with strings of the same size $n + nm$ from a random function R. Obviously

$$\Pr[G_0 \Rightarrow 1] \leq \Pr[G_1 \Rightarrow 1] + \mathsf{Adv}^{\text{prg}}_{G}(\mathcal{B}) \tag{1}$$

where $\mathsf{Adv}^{\text{prg}}_{G}(\mathcal{B})$ is the advantage of an adversary $\mathcal{B}$ to distinguish truly random string from R from pseudorandom strings from G making $q_{\mathsf{PRG}} = q_{\mathsf{Enc}} + q_{\mathsf{Dec}} + q_{\mathsf{Challenge}}$ queries in time complexity t.

We number the pairwise different nonces $N_j, j = 1 \ldots q$ as they appear in the oracle queries. We let J be the index of the nonces appeared in $\mathsf{Challenge}$ query and made win switch to true. Then we have that:

$$\Pr[G_1 \Rightarrow 1] = \sum_{j=1}^{q} \Pr[G_1 \Rightarrow 1 : \ J = j] \tag{2}$$

Recall that $\mathcal{A}$ wins the MO-nCTXT2 game only if it manages to present a tuple (H, N, C_1, C_2) to the $\mathsf{Challenge}$ oracle without having queried the Dec oracle to avoid the trivial attack given by [GLR17]. For each $N_j, j \in [1, \ldots q]$, we define adversaries $\mathcal{C}_j$ against the universal unforgeability on chosen messages against the collision resistance pseudorandom function F^{cr}, which is keyed by $P_0 = G(\mathsf{k}, N_j, n)$. We make the game abort if N_j is queried to the Dec oracle. Thus:

$$\Pr[G_1 \Rightarrow 1] \leq \mathsf{Adv}^{\text{uf}-\text{cma}}_{\mathsf{F}^{cr}}(\mathcal{C}_j) \tag{3}$$

Finally from $(1), (2), (3)$ accumulating the distinguishing probabilities of $\mathcal{A}$ against the MO-nCTXT2 game we have:

$$\mathsf{Adv}^{\text{mo-nctxt2}}_{\mathsf{CEP2}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{prg}}_{G}(\mathcal{B}) + \sum_{j=1}^{q} \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{F}^{cr}}(\mathcal{C}_j)$$

□

Theorem 2 (CEP2 Confidentiality). *Let* $\mathsf{CEP2} = \mathsf{CEP}[\mathsf{F}, G]$ *be a message franking scheme and a* MO-nRoR *adversary* $\mathcal{A}$ *making at most* $q = (q_{\mathsf{Enc}}, q_{\mathsf{Dec}}, q_{\mathsf{Challenge}})$ *queries in time complexity* t*. There exist adversaries* $\mathcal{B}, \mathcal{C}$ *making at most* $q_{\mathsf{PRG}} = q_{\mathsf{Enc}} + q_{\mathsf{Dec}} + q_{\mathsf{Challenge}}$ *and* $q_{\mathsf{PRF}} = q_{\mathsf{Enc}} + q_{\mathsf{Dec}} + q_{\mathsf{Challenge}}$ *queries in time complexity* t *accordingly such that:*

$$\mathsf{Adv}^{\mathrm{mo-nror}}_{\mathsf{CEP2}}(\mathcal{A}) \leq 2 \cdot \mathsf{Adv}^{\text{mo-nctxt2}}_{\mathsf{CEP2}}(\mathcal{B}) + \mathsf{Adv}^{\mathrm{prf}}_{\mathsf{F}^{cr}}(\mathcal{C})$$

Proof. Let G_0 be the MO-nREAL game. We modify G_0 in G_1, by introducing y, l and win as in the MO-nCTXT2 game and by making G_1 abort if $\mathsf{win} = \mathsf{true}$. Then we have:

$$\mathsf{Adv}^{\mathrm{mo-nror}}_{\mathsf{CEP2}}(\mathcal{A}) \leq \mathsf{Adv}_{G_1}(\mathcal{A}) + 2 \cdot \mathsf{Adv}^{\text{mo-nctxt2}}_{\mathsf{CEP2}}(\mathcal{B})$$

In G_1 we are ensured that the nonce N submitted to the $\mathsf{Challenge}$ oracle is never submitted to the Dec oracle but to return $\perp$. Then we can reduce to the PRF game such that $\mathsf{Adv}_{G_1}(\mathcal{A}) = \mathsf{Adv}^{\mathrm{prf}}_{\mathsf{F}^{cr}}(\mathcal{C})$. Finally it holds that:

$$\mathsf{Adv}^{\mathrm{mo-nror}}_{\mathsf{CEP2}}(\mathcal{A}) \leq 2 \cdot \mathsf{Adv}^{\text{mo-nctxt2}}_{\mathsf{CEP2}}(\mathcal{B}) + \mathsf{Adv}^{\mathrm{prf}}_{\mathsf{F}^{cr}}(\mathcal{C})$$

□

Sender binding s-BIND follows trivially from CEP2-Verify algorithm as long as the router executes honestly the verification algorithm and $\mathcal{R}$ runs decryption accordingly. Namely $\mathcal{R}$ checks whether $C_2 = C_2'$ and the same check is performed by the router.

Receiver binding r-BIND for CEP2 lies on the collision resistance property of F^{cr}, similarly with CEP:

Theorem 3 (CEP2 Receiver Binding). *Let* CEP2 *be a message franking scheme and* $\mathcal{A}$ *an adversary against* r-BIND *with time complexity* t*. Then, there exists an adversary* $\mathcal{B}$ *finding a collision of* F^{cr} *with time complexity* t*:*

$$\mathsf{Adv}^{\mathrm{r-bind}}_{\mathsf{CEP2}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathrm{cr}}_{\mathsf{F}^{cr}}(\mathcal{B})$$

Proof (Sketch). $\mathcal{B}$ runs $\mathcal{A}$ until the latter outputs a tuple $((H, M, \mathsf{sk}_f), (H', M', \mathsf{sk}_f'), C_2)$, whereby the r-BIND game outputs 1. That is, $\mathsf{Verify}(H, M, \mathsf{sk}_f, C_2) = \mathsf{Verify}(H', M', \mathsf{sk}_f, C_2) = 1 \Rightarrow \mathsf{F}^{cr}_{\mathsf{sk}_f}(H \| M)' = \mathsf{F}^{cr}_{\mathsf{sk}_f}(H' \| M')$, thus a valid collision of F^{cr}. □

4 PMF: Private Message Franking

4.1 Privacy Leakage with CEP

CEP [GLR17] introduces an increased leakage of confidentiality for the non-abusive message due to the way the protocol and the security games treat the entire message as a singleton object. Namely, each time a benign $\mathcal{R}$ opens an

abusive message $\tilde{M}$ all the blocks $\{b_i\}_{i=1}^{|M|/n}$ are revealed to the router. A single message though consists of multiple blocks of equal length n. Some α blocks of them may render the entire message abusive, but the rest β, $\alpha + \beta = |M|/n$ may need to be kept secret and not be open to the router. The current CEP construction does not treat the message M as a set of blocks, rather operates during the opening procedure at the entire M.

In our approach we first extend the current model for message franking in order to adhere to the partial opening property, which protects the non-abusive blocks from the abusive ones in one message M. Namely, we introduce a predicate relationship $R()$, which takes as input a message M and outputs 1 whenever the message contains abusive blocks and 0 otherwise. We also separate from the decryption function Dec the opening functionality in a separate algorithm Proof, which outputs a proof Π, demonstrating to a router that a message M is considered as abusive, due to some blocks, which are opened to the router. The latter verifies the proof calling the Verify algorithm which takes as input the proof Π. More formally we define our new syntactical model for Committing Nonce based Authenticated Encryption with Partial Opening in the following subsection.

4.2 Committing Nonce based Authenticated Encryption with Partial Opening(CEPO)

A CEPO scheme consists of five algorithms (KGen, Enc, Dec, Proof, Verify), associated with a message space $\mathcal{M} \in \Sigma^*$, a key space $\mathcal{K} \in \Sigma^*$, a nonce space $\mathcal{N} \in \Sigma^*$, a header space $\mathcal{H} \in \Sigma^*$, a ciphertext space $\mathcal{C} \in \Sigma^*$, an opening space $\mathcal{O} \in \Sigma^*$, a franking space $\mathcal{T} \in \Sigma^*$ and a proof space $\mathcal{P} \in \Sigma^*$. The five algorithms are defined as follows:

- $\mathsf{k} \leftarrow\!\!\$\ \mathsf{KGen}(1^\lambda)$: A randomized algorithm, which outputs a secret key $\mathsf{k} \in \mathcal{K}$, on input a security parameter λ in its unary form.
- $(C_1, C_2) \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{k}, H, N, M)$: The encryption algorithm, which is deterministic, takes as input a key, a header, a nonce and a message $(\mathsf{k}, H, N, M) \in (\Sigma^*)^4$ and outputs $(C_1, C_2) \in \mathcal{C} \times \mathcal{T}$ or $\perp$. C_1 will be usually referred to as the ciphertext and C_2 as the commitment.
- $(\mathsf{sk}_f, M) \leftarrow \mathsf{Dec}(\mathsf{k}, H, N, C_1, C_2)$: The decryption algorithm Dec is a deterministic algorithm, which takes as input $(\mathsf{k}, H, N, C_1, C_2) \in (\Sigma^*)^5$ and outputs a message $M \in \mathcal{M}$ with an opening key $\mathsf{sk}_f \in O$, or $\perp$.
- $\Pi \leftarrow \mathsf{Proof}(R, H, M, \mathsf{sk}_f)$: This is a deterministic algorithm, which takes as input $(R, M, H, \mathsf{sk}_f) \in (\Sigma^*)^4$ and outputs a proof $\Pi \in \mathcal{P}$, which demonstrates correctness of the predicate R on input the message M. The predicate R is defined as follows:

$$R(M) = \begin{cases} 1, B = (i, \ldots, j) & \text{if } \exists i, j \in [1 \ldots m] \text{ s.t. } b_i, \ldots b_j = x_i, \ldots x_j, \\ & x_i \in \{0,1\}^n \\ 0, & \text{otherwise} \end{cases}$$

where $b_1 \dots b_m$ are the blocks of the message M. The predicate returns 1 and a set B of the indeces $i, \dots j$, whenever some blocks of the message $b_i, \dots b_j$ equal to some specific bitstrings $x_i, \dots x_j, x_i \in \{0,1\}^n$ which are regarded as abusive. What is flagged as abusive is inherently implied in the protocol. It is up to the choice of the receiver what it will be considered as abusive and what not. A malicious receiver, who always opens blocks of a message to the router, even if these are not flagged as abusive by the router is not captured in the model, as this seems impossible to be enforced technically.

– $0, 1 \leftarrow \mathsf{Verify}(H, \Pi, \mathsf{fk})$: This deterministic algorithm takes as input $(H, \Pi, \mathsf{fk}) \in (\mathcal{H} \times \mathcal{P} \times \mathcal{K})$ and outputs 1 if verification is successful and 0 otherwise.

We write $\mathsf{Enc}_\mathsf{k}(\cdot,\cdot,\cdot)[1]$ and $\mathsf{Enc}_\mathsf{k}(\cdot,\cdot,\cdot)[2]$ to denote C_1 and C_2 and accordingly $\mathsf{Dec}_\mathsf{k}(\cdot,\cdot,\cdot,\cdot)[1], \mathsf{Dec}_\mathsf{k}(\cdot,\cdot,\cdot,\cdot)[2]$ for sk_f and M.

A CEPO is correct if it adheres to 1) *decryption correctness* as in correctness for encryption schemes: $\forall(\mathsf{k}, H, N, M) \in (\mathcal{K} \times \mathcal{H} \times \mathcal{N} \times \mathcal{M})$ it is true that:

$$\Pr[\underbrace{\mathsf{Dec}(\mathsf{k}, H, N, \underbrace{\mathsf{Enc}_\mathsf{k}(H, N, M)}_{C_1, C_2})}_{M}[2] = M] = 1$$

and 2) commitment correctness: if $\forall(\mathsf{k}, H, M) \in (\mathcal{K} \times \mathcal{H} \times \mathcal{M})$ it is true that:

$$\Pr[\mathsf{Verify}(H, \overbrace{\mathsf{Proof}(R, H, M, \mathsf{sk}_f)}^{\Pi}, \mathsf{fk}) = 1] = 1$$

where $\mathsf{sk}_f = \mathsf{Dec}_\mathsf{k}(H, N, \mathsf{Enc}_\mathsf{k}(H, N, M))[1]$.

Throughout the model for message franking as first captured [GLR17] and instantiated with the CEP protocol, the tasks performed by Facebook are omitted in the model and during costs analysis. That is, the signing operation performed by Facebook on C_2 and on the metadata $md = \mathcal{S} \| \mathcal{R} \| timestamp$ are discarded in the protocol. We conjecture that this is due to the fact that Facebook at the Verify algorithm always acts honestly and the cost for one extra signing and verification operation is negligible. In our two protocols we enhance the model to be more accurate with the existing API of Facebook, including an algorithm called Process, which illustrates the tasks performed by a router when receiving C_1, C_2 by the sender $\mathcal{S}$.

With our two protocols CEP-AOP1 and CEP-AOP2 for message franking with after opening privacy we enhance the privacy of the current message franking protocols with *after opening privacy*: The message is not treated as a singleton object, rather it is split in blocks and only the abusive blocks are opened by the receiver $\mathcal{R}$ to the router. We present in the next section the stronger privacy guarantee modeled with cryptographic game.

4.3 After Opening Privacy

In order to enhance the security guarantees of messaging protocols with after opening privacy we introduce a game based definition for multi opening indistinguishable partial openings (MO-IND-PO). Intuitively that security definition

guarantees the confidentiality of the closed blocks: those which did not open to the router by the receiver $\mathcal{R}$, when the latter blacklists a message M as abusive due to some abusive blocks.

The game is presented in Figure 8. We omit the explanation of the Enc and Dec oracles since these are replicated directly from the confidentiality game of the CEP scheme [GLR17]. Apart from the Enc and Dec oracles, $\mathcal{A}$ has access to the Proof oracle. That oracle takes as input the partial opening predicate function R. The oracle first checks if the challenged pair of messages result in the same predicate R evaluation to capture trivial attacks, whereby the adversary $\mathcal{A}$ guesses correctly with probability 1 during the challenge. Namely $\mathcal{A}$ can open some blocks of the message, which evaluates correctly the predicate R and verify with the opening key (because the predicate of that message equals 1) which message has been encrypted by the Challenge oracle. If the predicate evaluation over the challenged messages M_0 and M_1 is equal then Proof oracle proceeds with the decryption of the input tuple (H, N, C_1, C_2) to learn (sk_f, M) and then runs the Proof algorithm on input (R, H, M, sk_f) to learn the proof Π. Finally it forwards Π to $\mathcal{A}$.

When $\mathcal{A}$ decides to get challenged, it calls the Challenge oracle on input (H, N, M_0, M_1, R), under the condition that the nonce N has not been queried before, $|M_0| = |M_1|$, chall is empty and messages evaluate to the same output on the predicate R. The challenger also checks whether the same nonce has been given as input at any call to the Enc oracle and halts the game if so to avoid distinguishing attacks on the underlying authenticated encryption scheme. Then it encrypts M_b and returns to $\mathcal{A}$ the ciphertext C_1 with the tag C_2. In the indistinguishable flavor we say that the advantage of an adversary $\mathcal{A}$ while playing the MO-IND-PO is the probability of $\mathcal{A}$ to output $b' = b$ at the end of the game:

$$\mathsf{Adv}^{\mathrm{mo-ind-po}}_{\mathsf{CEPO}}(\mathcal{A}) = |\Pr[\text{MO-IND-PO}(0) \Rightarrow 1] - \Pr[\text{MO-IND-PO}(1) \Rightarrow 1]|$$

After enhancing the privacy requirements of a message franking protocol with the after opening privacy notion as formalized in the previous section we can embark on our solution ideas. We first give a naive solution, which hides the non-abusive messages, but introduces an increased communication complexity. We call this protocol CEP-AOP1. We then present our optimized protocol CEP-AOP2, and analyze its security in a formal way.

5 CEP-AOP1

5.1 Description

We consider as the basis for our message franking protocol with after opening privacy the CEP construction [GLR17], which achieves the multi-opening confidentiality and integrity notions and needs less passes over the message, compared

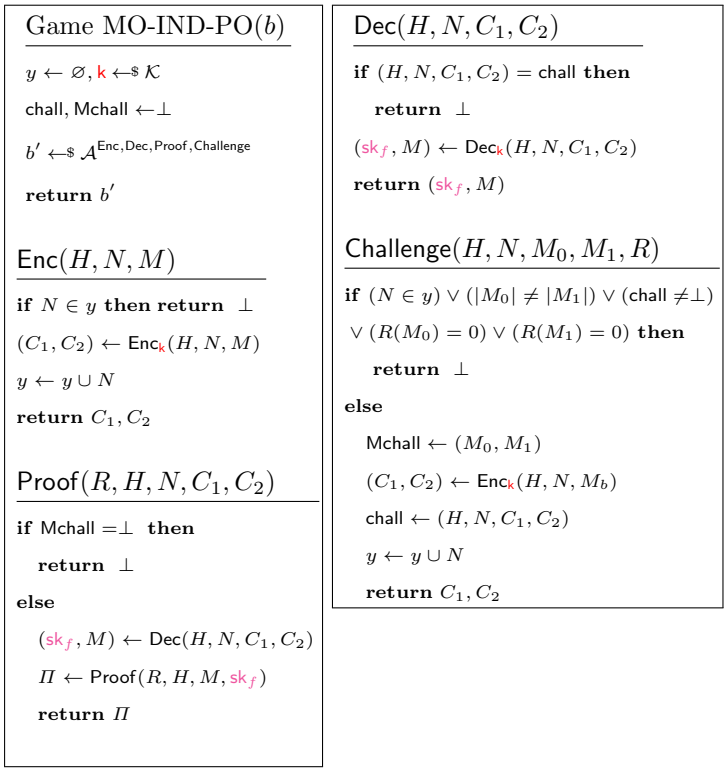

Fig. 8: Game MO-IND-PO

with the compositional designs of Encode-then-Encipher [BR00], Encrypt-then-Mac [BR00] and Mac-then-Encrypt. The privacy leakage of the CEP protocol occurs during the opening phase. The receiver of a message thinking that it violates its abusiveness limits, reports it in a verifiable manner to the router. Namely, the router is exposed to the CEP-Verify$(H, M, \mathsf{sk}_f, C_2)$ algorithm, which takes as input the entire message M with the authentication tag key sk_f and the commitment C_2. The challenge is dual: First, the new protocol has to maintain the receiver binding property for the abusive blocks of the message such as it cannot faulty blame the sender for message that it did not send. In parallel, the router after receiving the secret authentication tag key sk_f should not be able to compromise the blocks which have not be opened by $\mathcal{R}$, while verifying the integrity of the claimed as abusive by $\mathcal{R}$ blocks.

The shared encryption key k is never opened to the router. Consequently, encryption and decryption algorithms are not altered. Our first solution principle which is described in Figure 9 works as follows:

CEP-AOP1-Enc$_{\mathsf{k}}(H, N, M)$:

$m \leftarrow |M|/n$

parse M **as** $m_i[0 \dots m-1]$

$P \leftarrow G(\mathsf{k}, N, 2mn)$

$C_1 \leftarrow (P_m \| \dots \| P_{2m-1} \oplus M)$

for $i = 0 \dots m-1$ **do** :

  $c_2^i \leftarrow \mathsf{F}_{P_i}^{cr}(H \| m_i)$

return $(C_1, C_2 = c_2^i[0 \dots m-1])$

CEP-AOP1-Process$(\mathcal{S}, \mathcal{R}, C_1, C_2, \mathsf{fk})$:

parse C_2 **as** $c_2^i[0 \dots m-1]$

$md \leftarrow \mathcal{S} \| \mathcal{R} \| \text{time}$

for $i = 0 \dots m-1$ **do** :

  $s_i \leftarrow c_2^i \| md$

  $a_i \leftarrow \mathsf{F}_{\mathsf{fk}}(s_i)$

return $(C_1, C_2 = c_2^i[0 \dots m-1], a_i[0 \dots m-1])$

CEP-AOP1-Dec$_{\mathsf{k}}(H, N, C_1, C_2)$:

$m \leftarrow |C_1|/n$

parse C_2 **as** $c_2^i[0 \dots m-1]$

$P \leftarrow G(\mathsf{k}, N, 2nm)$

$M \leftarrow (P_m \| \dots \| P_{2m-1} \oplus C_1)$

parse M **as** $m_i[0 \dots m-1]$

for $i = 0 \dots m-1$ **do** :

  $c_2'^i \leftarrow \mathsf{F}_{P_i}^{cr}(H \| m_i)$

  if $c_2'^i \neq c_2^i$ **then return** $\perp$

return $(M, \mathsf{sk}_f = P_i[0 \dots m-1])$

CEP-AOP1-Proof(R, H, M, sk_f):

if $R(M) = 0$ **then return** $\perp$

parse sk_f **as** $P_i[0 \dots m-1]$

parse M **as** $b_i[0 \dots m-1]$

return $\Pi = (H, \{b_i\}, P_i, a_i, i \in B)$

CEP-AOP1-Verify(H, Π, fk):

parse Π **as** $H, \{b_i\}, P_i, a_i, i \in B$

for $i \in B$ **do** :

  $c_2'^i \leftarrow \mathsf{F}_{P_i}^{cr}(H \| b_i)$

  $a_i' \leftarrow \mathsf{F}_{\mathsf{fk}}(c_2'^i \| md)$

  if $a_i' \neq a_i$ **then return** $\perp$

return 1

Fig. 9: CEP-AOP1 algorithms

During encryption the sender $\mathcal{S}$ calls the nonce-based pseudorandom generator G[2] with desired output size $2mn$, where $m = |M|/n$, for a block size of n bits. The encryption as with the CEP scheme operates as a xor based one time pad.

[2] We use the same naming with [GLR17] for the pseudorandom generator G introduced as a nonce-based taking as input the nonce N, however the model is reminiscent to pseudorandom generators with input as first introduced in [BH05] and later enhanced in [DPR+13] with stronger security guarantee: robustness.

The first blocks of randomness $P_i, i \in [0 \dots m-1]$ are used to key the collision resistant PRF F^{cr}. P_i denotes the i^{th} block of randomness of size n bits. The main difference with the CEP [GLR17] scheme is that there is one tag per block instead for one tag for the entire message in order to adhere to *after opening privacy* of the non-abusive blocks. $\mathcal{S}$ forwards the encrypted ciphertect C_1 and the commitment $C_2 = c_2^i[0 \dots m-1]$ to the router. The latter iterates over all the tags, and computes one authentication tag a_i per block using its own secret key fk. Finally, the router forwards to $\mathcal{R}$ $C_1, C_2 = c_2^i[0 \dots m-1], a_i[0 \dots m-1]$.

When the receiver $\mathcal{R}$ gets C_1, C_2 calls the $\mathsf{CEP\text{-}AOP1\text{-}Dec_k}(H, N, C_1, C_2)$ algorithm to decrypt the message and check its integrity. It first parses the ciphertext C_1 and the commitment C_2 as $c_2^i[0 \dots m-1]$. It then calls the nonce-based pseudorandom generator on input the common agreed key k to produce the pad P of size $2mn$ bits. Afterwards it parses the decrypted message M in message blocks $m_i[0 \dots m-1]$ and recomputes the tags $c_2'^i$, for $i \in [0 \dots m-1]$ using as keys the pads P_{m+i} for the $\mathsf{PRF} : \mathsf{F}^{cr}_{P_{m+i}}$.

$\mathcal{R}$ calls the $\mathsf{CEP\text{-}AOP1\text{-}Proof}(R, H, M, C_2, \mathsf{sk}_f)$ algorithm in order to provide a proof to the router, demonstrating that some message M sent by the sender $\mathcal{S}$ contains abusive blocks $b_i, i \in B$ indexed in the set B. The algorithm outputs a proof consisting of the opening keys $P_i, i \in B$ only for the abusive blockss $b_i, i \in B$. Finally the router verifies the correctness of the proof by calling the $\mathsf{CEP\text{-}AOP1\text{-}Verify}(H, \Pi, C_2)$ algorithm and checks whether the tags of abusive blocks are consistent, using the opening keys $P_i, i \in B$ to re-evaluate the collision resistant $\mathsf{PRF} : \mathsf{F}^{cr}$.

5.2 Security Analysis

Theorem 4 ($\mathsf{CEP\text{-}AOP1}$ Integrity). *Let* $\mathsf{CEP\text{-}AOP1}[\mathsf{F}, G]$ *be a* $\mathsf{CEP\text{-}AOP}$ *scheme and a* $MO - nCTXT$ *adversary* $\mathcal{A}$ *making at most* q *queries. Then there exist adversaries* $\mathcal{B}$ *and* $\mathcal{C}$ *making each* $q_{\mathsf{PRG}} = q_{\mathsf{Enc}} + q_{\mathsf{Dec}} + q_{\mathsf{Challenge}}$ *and* $q_{\mathsf{PRF}} = q_{\mathsf{Enc}} + q_{\mathsf{Dec}} + q_{\mathsf{Challenge}}$ *queries in time complexity* t *such that:*

$$\mathsf{Adv}^{\text{mo-nctxt2}}_{\mathsf{CEP\text{-}AOP1}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{prg}}_{G}(\mathcal{B}) + \sum_{j=1}^{m \cdot q} \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{F}^{cr}}(\mathcal{C}_j)$$

Proof. Similarly with the integrity proof for the $\mathsf{CEP2}$ protocol we assume without loss of generality that all queries (H, N, C_1, C_2) to the Dec oracle are in the y list or that $N \in l$. Otherwise we can use the $\mathsf{Challenge}$ oracle. The game halts also as soons as $\mathsf{win} = \mathsf{true}$. Let G_0 be the original game MO-nCTXT2 and G_1 is equivalent with G_0 except that calls G are replaced with strings of the same size $2nm$ from a random function R. Then, it holds:

$$\Pr[G_0 \Rightarrow 1] \leq \Pr[G_1 \Rightarrow 1] + \mathsf{Adv}^{\text{prg}}_{G}(\mathcal{B}) \tag{4}$$

where $\mathsf{Adv}^{\text{prg}}_{G}(\mathcal{B})$ is the advantage of an adversary $\mathcal{B}$ to distinguish truly random string from R from pseudorandom strings from G making $q_{\mathsf{PRG}} = q_{\mathsf{Enc}} + q_{\mathsf{Dec}} + q_{\mathsf{Challenge}}$ queries in time complexity t.

We enumerate the pairwise different nonces $N_j, j = 1 \dots q \cdot m$ as they appear in the oracle queries. We let J be the index of the nonces appeared in Challenge query and made win switch to true. Notice that compared with CEP2, enumeration of nonces goes for each different key stream $P_i, i = 1 \dots m$, for each block b_i. We let q queries for each different key stream. Then we have that:

$$\Pr[G_1 \Rightarrow 1] = \sum_{j=1}^{q \cdot m} \Pr[G_1 \Rightarrow 1 : \ J = j] \tag{5}$$

$\mathcal{A}$ wins the MO-nCTXT2 game only if it manages to present a tuple (H, N, C_1, C_2) to the Challenge oracle without having queried the Dec oracle to avoid the trivial attack given by [GLR17]. For each $N_j, j \in [1, \dots q]$, we define adversaries $\mathcal{C}_j$ against the universal unforgeability on chosen messages against the collision resistance pseudorandom function F^{cr} keyed by $P_j, j \in [1 \dots m]$. We make $\mathcal{C}_j$ abort if N_j is queried to the Dec oracle. Thus:

$$\Pr[G_1 \Rightarrow 1] \leq \mathsf{Adv}^{\mathrm{uf-cma}}_{\mathsf{F}^{cr}}(\mathcal{C}_j) \tag{6}$$

Finally from (4), (5), (6) and accumulating the distinguishing probabilities of $\mathcal{A}$ against the MO-nCTXT2 game we have:

$$\mathsf{Adv}^{\text{mo-nctxt2}}_{\text{CEP-AOP1}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{prg}}_{G}(\mathcal{B}) + \sum_{j=1}^{m \cdot q} \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{F}^{cr}}(\mathcal{C}_j).$$

□

Sender binding is guaranteed as long as decryption algorithm Dec decrypts correctly: it outputs the correct message M or $\perp$ when there is an error, and Verify run by an honest router.

Theorem 5 (CEP-AOP1 Confidentiality). *Let* CEP-AOP1$[\mathsf{F}, G]$ *be a* CEP-AOP *scheme and a* MO-nRoR *adversary* $\mathcal{A}$ *making at most* $q = q_{\mathsf{Enc}}, q_{\mathsf{Dec}}, q_{\mathsf{Challenge}}$ *queries with time complexity* t*. Then there exist adversaries* $\mathcal{B}$ *making* $q_{\mathsf{PRG}} = q_{\mathsf{Enc}} + q_{\mathsf{Dec}} + q_{\mathsf{Challenge}}$ *and* $\mathcal{C}$ *making* $q_{\mathsf{PRG}} = q_{\mathsf{Enc}} + q_{\mathsf{Dec}} + q_{\mathsf{Challenge}}$ *queries in time complexity* t *each, such that:*

$$\mathsf{Adv}^{\mathrm{mo-nror}}_{\text{CEP-AOP1}}(\mathcal{A}) \leq 2 \cdot \mathsf{Adv}^{\text{mo-nctxt2}}_{\text{CEP-AOP1}}(\mathcal{B}) + m \cdot \mathsf{Adv}^{\mathrm{prf}}_{\mathsf{F}^{cr}}(\mathcal{C})$$

Proof. Let G_0 be the MO-nREAL game. We change G_0 in G_1 as with the confidentiality proof for CEP2. We introducing y, l lists and win variable as in the MO-nCTXT2 game and by making G_1 abort if win = true. Then it holds that:

$$\mathsf{Adv}^{\mathrm{mo-nror}}_{\text{CEP-AOP1}}(\mathcal{A}) \leq \mathsf{Adv}_{G_1}(\mathcal{A}) + 2 \cdot \mathsf{Adv}^{\text{mo-nctxt2}}_{\text{CEP-AOP1}}(\mathcal{B})$$

In G_1 we are ensured that the nonce N submitted to the Challenge oracle is never submitted to the Dec oracle but to return $\perp$. Then we can reduce to the PRF game such that $\mathsf{Adv}_{G_1}(\mathcal{A}) = m \cdot \mathsf{Adv}^{\mathrm{prf}}_{\mathsf{F}^{cr}}(\mathcal{C})$.

Finally it holds that:

$$\mathsf{Adv}^{\text{mo}-\text{nror}}_{\text{CEP-AOP1}}(\mathcal{A}) \leq 2 \cdot \mathsf{Adv}^{\text{mo-nctxt2}}_{\text{CEP-AOP1}}(\mathcal{B}) + m \cdot \mathsf{Adv}^{\text{prf}}_{\mathsf{F}^{cr}}(\mathcal{C})$$

□

Theorem 6 (CEP-AOP1 After Opening Privacy). *Let* CEP-AOP1$[\mathsf{F}, G]$ *be a* CEP-AOP *scheme and a* MO-IND-PO *adversary* $\mathcal{A}$ *making* $q = (q_{\mathsf{Enc}}, q_{\mathsf{Dec}}, q_{\mathsf{Proof}}, q_{\mathsf{Challenge}})$ *queries in time complexity* t*. Then there exist adversaries* $\mathcal{B}$ *and* $\mathcal{C}$ *making each* $q_{\mathsf{PRG}} = q_{\mathsf{Enc}} + q_{\mathsf{Dec}} + q_{\mathsf{Challenge}} + q_{\mathsf{Proof}}$ *and* $q_{\mathsf{PRF}} = q_{\mathsf{Enc}} + q_{\mathsf{Dec}} + q_{\mathsf{Challenge}} + q_{\mathsf{Proof}}$ *queries in time complexity* t *such that:*

$$\mathsf{Adv}^{\text{mo-ind-po}}_{\text{CEP-AOP1}}(\mathcal{A}) \leq \mathsf{Adv}^{\text{mo}-\text{nror}}_{\text{CEP-AOP1}}(\mathcal{B}) + m \cdot \mathsf{Adv}^{\text{prf}}_{\mathsf{F}^{cr}}(\mathcal{C})$$

Proof. Let game G_0 be identical with the MO-IND-PO game.

In game G_1 we substitute y with y_0 and we introduce y_1 similarly with the confidentiality game MO-nRoR. Whenever MO-nRoR halts G_0 also halts. In the Challenge oracle $\mathcal{A}$ submits messages M_0 and M_1 such that $(N \notin y_0) \wedge (|M_0| = |M_1|) \wedge (\mathsf{chall} = \perp) \wedge (R(M_0) = 1) \wedge (R(M_1) = 1)$. Whenever $b = 0$ in the Challenge of G_1 the game returns to $\mathcal{A}$ $(C_1, C_2) \leftarrow \mathsf{Enc}_{\mathsf{k}}(H, N, M_0)$. When $b = 1$ G_1 runs $(C_1, C_2) \leftarrow \mathsf{Enc}_{\mathsf{k}}(H, N, \{0,1\}^{|M|})$. Thus:

$$\Pr[G_0 \Rightarrow 1] \leq \Pr[G_1 \Rightarrow 1] + \mathsf{Adv}^{\text{mo}-\text{nror}}_{\text{CEP-AOP1}}(\mathcal{B})$$

$\mathcal{A}$ can also win the MO-IND-PO game if she manages to forge C_2 in order to issue a $\mathsf{chal} = (H, N, C_1, C_2')$ tuple in the Dec oracle, bypass the check, decrypt the chal query and distinguish with non negligible probability.

Finally it holds:

$$\mathsf{Adv}^{\text{mo-ind-po}}_{\text{CEP-AOP1}}(\mathcal{A}) \leq \mathsf{Adv}^{\text{mo}-\text{nror}}_{\text{CEP-AOP1}}(\mathcal{B}) + m \cdot \mathsf{Adv}^{\text{prf}}_{\mathsf{F}^{cr}}(\mathcal{C})$$

□

Theorem 7 (CEP-AOP1 Receiver Binding). *Let* CEP-AOP1 *be a message franking scheme and* $\mathcal{A}$ *an adversary against* r-BIND *with time complexity* t*. Then, there exists an adversary* $\mathcal{B}$ *finding a collision of* F^{cr} *with time complexity* t*:*

$$\mathsf{Adv}^{\text{r}-\text{bind}}_{\text{CEP-AOP1}}(\mathcal{A}) \leq m \cdot \mathsf{Adv}^{\text{cr}}_{\mathsf{F}^{cr}}(\mathcal{B})$$

Proof (Sketch). $\mathcal{B}$ runs $\mathcal{A}$ until the latter outputs a tuple $\{((H, b_i, \mathsf{sk}_f), (H', b_i', \mathsf{sk}_f'), C_2)\}_{i \in B}$, whereby the r-BIND game outputs 1. That is, $\mathsf{Verify}(H, b_i, \mathsf{sk}_f, C_2) = \mathsf{Verify}(H', b_i', \mathsf{sk}_f, C_2) = 1 \Rightarrow \mathsf{F}^{cr}_{\mathsf{sk}_f}(H \| b_i)' = \mathsf{F}^{cr}_{\mathsf{sk}_f}(H' \| b_i')$ for some $i's \in B$, thus a valid collision of F^{cr}. The maximum value of B equals the number of blocks m, thus

$$\mathsf{Adv}^{\text{r}-\text{bind}}_{\text{CEP-AOP1}}(\mathcal{A}) \leq m \cdot \mathsf{Adv}^{\text{cr}}_{\mathsf{F}^{cr}}(\mathcal{B})$$

□

5.3 Shortcomings for CEP-AOP1

For each encrypted message the sender $\mathcal{S}$ is willing to send to the receiver $\mathcal{R}$ through the router, $\mathcal{S}$ has to call a pseudorandom generator G in order to extract $2nm$ bits of randomness. mn bits are used as a one time pad encryption of the m blocks of the message M and the rest mn are used to key the F^{cr} call for every block. Whenever $\mathcal{R}$ reports to the router the abusive blocks $b_i, i \in B$ it has to communicate the opening keys for the β PRF evaluations of the collision resistant pseudorandom function F^{cr}.

In CEP-AOP1, the cost of the router is not negligible: The router receives m tags $c_2^i, i \in [1 \dots m]$ for a single message M and has to sign with its private signing key fk all c_2^i tags and then verify the authentication tags on β presumably abusive blocks on top of the individual PRF evaluation with the opening keys, as received by $\mathcal{R}$.

In the following section we design and analyze our final protocol dubbed CEP-AOP2, which reduces the computation complexity at the router side and the communication cost between $\mathcal{S}$ and the router. Namely, the router is required to perform only one signing operation per message and still adhere to AOP, independently on the number of the blocks at each message M and $\mathcal{S}$ sends one commitment for all blocks. At the same time $\mathcal{R}$ can select the abusive blocks and keep the rest privy to the router, allowing him to verify only the validity of the abusive ones. For our protocol we exploit the Merkle Hash Tree (MHT), which acts as a signature over all the blocks, with efficient verification of a subset of leaves, without requiring the opening of the rest leaves for verification, thus adhering to AOP. Despite the increased computation cost the receiver is now charged for the computation of the Merkle tree, we conjecture that in a messaging application senders are dynamic but the router remains the same. As such, the overall computation workload cost per party is decreased.

6 CEP-AOP2

Before delving in the description of CEP-AOP2 protocol we put forth the Merkle Hash tree (MHT) data structure in the following subsection.

6.1 Merkle Trees

The use of Merkle Trees to authenticate streams of data has already been proposed by Merkle [Mer89]. A binary tree whereby the leaf nodes correspond to data and intermediate nodes keep digest thereof reduce the authentication procedure to logarithmic costs on the height of the tree and the size of data subsequently. Let $\mathtt{H}$ be a collision resistant hash function: $\mathtt{H}\{0,1\}^* \to \{0,1\}^\lambda$, mapping strings of arbitrary length to λ-bit strings. Assume a vector of data elements $\mathbf{l} = \{l_1, l_2, l_3, \dots, l_n\}$. The $\mathtt{MHT}(\mathbf{l})$ algorithm computes the Merkle tree for the data vector $\mathbf{l}$ and outputs its $\mathsf{rt}_\mathbf{l}$ (cf. figure 10-left). All leaf nodes correspond to the hash of each element $\mathtt{H}(l)$ and parent nodes are computed as the hash of the

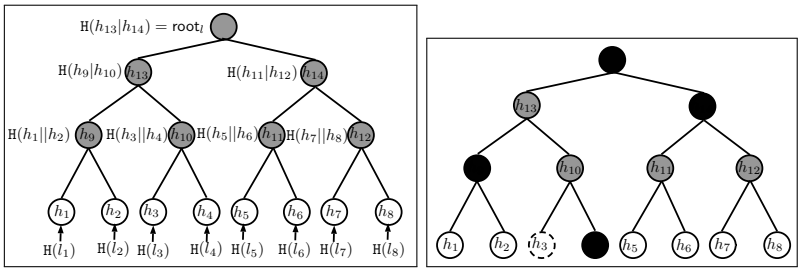

Fig. 10: Merkle Tree construction for the dataset $\mathbf{l} = l_1, l_2, l_3, l_4, l_5, l_6, l_7, l_8$. White nodes are leaf nodes. Grey nodes are intermediate-parent nodes.Black nodes consist the authentication path $\mathbf{ap}_{l_4} = \{h_4, h_9, h_{14}\}$ of the dashed line leaf node l_3. Note that there is no requirement the number of elements to be hashed to be a power of two. In that case the tree is not balanced.

concatenated children hashes. A prover who claims membership of data element l_x runs the $\texttt{ProveMT}(x, \mathbf{l})$ algorithm and sends the authentication path $\mathbf{ap}_{l_x}$ to the verifier. A verifier can check the correctness of the authentication path with respect to the membership of the element l_x in $\mathbf{l}$ by recomputing the Merkle tree based on the authentication path $\mathbf{ap}_{l_x}$ running the `CheckPath` algorithm. Finally it checks if the computed root $\texttt{H}' \stackrel{?}{=} \mathsf{rt}_l$. For example in figure 10-right $\mathbf{ap}_{l_3} = \{h_4, h_9, h_{14}\}$.

Security The security guarantee for a Merkle tree hash algorithm demonstrates that it is impossible for an adversary to forge a Merkle Tree Hash $\mathsf{rt}_l \leftarrow \texttt{MHT}(\mathbf{l})$ claiming valid membership proof for elements which do not belong in $\mathbf{l}$.

Definition 1. *If* $\texttt{H}$ *is a collision resistant hash function then the* $\texttt{MHT}(\mathbf{l})$ *hash algorithm outputs a merkle hash tree for the dataset* $\mathbf{l}$ *which is also collision resistant, i.e: an adversary* $\mathcal{A}$ *can claim to authenticate an element* $x \ni \mathbf{l}$ *by generating the same* $\mathsf{rt}_l \leftarrow \texttt{MHT}(\mathbf{l})$ *algorithm with negligible probability.*

6.2 Description

CEP-AOP2 (cf. Figure 11) operates as follows. Similarly with CEP-AOP1 the sender $\mathcal{S}$ encrypts its message M with the $\mathsf{CEP\text{-}AOP2\text{-}Enc}_\mathsf{k}(H, N, M)$ algorithm by choosing a sequence of $2nm$ random blocks. The first nm bits are used to encrypt m blocks of size n bits each. The rest are used to key a collision resistance $\mathsf{PRF}, \mathsf{F}^{cr}$. In contrast with CEP-AOP1, CEP-AOP2 forwards to the router C_1 and the root rt_{C_2} of a Merkle tree constructed over the tags $c_2^i[0 \ldots m-1]$. That is, as leaves we consider the evaluation of a PRF on each message block with different keys and the tree is constructed using a collision resistant hash function $\texttt{H}$. That drastically reduces the router costs as it only tags with its secret

CEP-AOP2-Enc$_{\mathsf{k}}(H, N, M)$:

$m \leftarrow |M|/n$

parse M **as** $m_i[0 \dots m-1]$

$P \leftarrow G(\mathsf{k}, N, 2mn)$

$C_1 \leftarrow (P_m \| \dots \| P_{2m-1} \oplus M)$

for $i = 0 \dots m-1$ **do** :

$c_2^i \leftarrow \mathsf{F}^{cr}_{P_{m+i}}(H \| m_i \| i)$

$\mathsf{rt}_{C_2} \leftarrow \mathsf{MHT}(C_2 = c_2^i, i \in [0 \dots m-1])$

return $(C_1, C_2 = \mathsf{rt}_{C_2})$

CEP-AOP2-Process$(\mathcal{S}, \mathcal{R}, C_2, \mathsf{fk})$:

parse C_2 **as** rt_{C_2}

$md \leftarrow \mathcal{S} \| \mathcal{R} \| \text{time}$

$s \leftarrow \mathsf{rt}_{C_2} \| md$

$a \leftarrow \mathsf{F}_{\mathsf{fk}}(s)$

return a

CEP-AOP2-Dec$_{\mathsf{k}}(H, N, C_1, C_2)$:

$m \leftarrow |C_1|/n$

parse C_2 **as** rt_{C_2}

$P \leftarrow G(\mathsf{k}, N, 2nm)$

$M \leftarrow (P_m \| \dots \| P_{2m-1} \oplus C_1)$

parse M **as** $m_i[0 \dots m-1]$

for $i = 0 \dots m-1$ **do** :

$c_2'^i \leftarrow \mathsf{F}^{cr}_{P_{m+i}}(H \| m_i \| i)$

$\mathsf{rt}'_{C_2} \leftarrow \mathsf{MHT}(C_2 = c_2'^i[i \dots m])$

if $\mathsf{rt}_{C_2} \neq \mathsf{rt}'_{C_2}$ **then return** $\perp$

return $(M, \mathsf{sk}_f = \mathsf{F}^{cr}_{P_{m+i}})$

CEP-AOP2-Proof(R, H, M, sk_f):

if $R(M) = 0$ **then return** $\perp$

parse sk_f **as** $P_i[0 \dots m-1]$

parse M **as** $b_i[0 \dots m-1]$

ap $\leftarrow \mathtt{ProveMT}(c_2^i[i \in B], C_2)$

return $\Pi = (H, \mathbf{ap}, \{b_i\}, P_i, a, i \in B)$

CEP-AOP2-Verify(H, Π, fk):

parse Π **as** $(H, \mathbf{ap}, \{b_i\}, P_i, a, i \in B)$

for $i \in B$ **do** :

$c_2'^i \leftarrow \mathsf{F}^{cr}_{P_{m+i}}(H \| b_i)$

if $\mathtt{CheckPath}(\mathbf{ap}, \mathsf{rt}_{C_2}, c_2'^i, i \in B) \neq 1$ **then return** $\perp$

$a' \leftarrow \mathsf{F}_{\mathsf{fk}}(\mathsf{rt}_{C_2} \| md)$

if $a' \neq a$ **then return** $\perp$

return 1

Fig. 11: CEP-AOP2 algorithms

key only one element at the Process algorithm: the root rt_{C_2} of the Merkle tree, which authenticates the tags $c_2^i[0 \dots m-1]$.

During decryption the CEP-AOP2-Dec$_{\mathsf{k}}(H, N, C_1, C_2)$ algorithm reproduces the same sequence of random blocks and uses them to decrypt C_1 and to reconstruct the Merkle tree. If the computed new root rt'_{C_2} agrees with rt_{C_2}, $\mathcal{R}$

accepts the message M as valid, otherwise it halts the procedure. If $\mathcal{R}$ considers some of the blocks $b_i[i \in B]$ as abusive, then it forwards them to the router, along with the opening keys $\mathsf{sk}_f = P_{m+i}, i \in B$ and the sibling path **ap** corresponding to the abusive block indexes. The router in turn, with the CEP-AOP2-Verify$(H, b_i, \mathsf{sk}_f, C_2)$ algorithm reevaluates the PRF using the opening keys and verifies that those leaves with the sibling path **ap** correctly verify the Merkle tree.

6.3 Security Analysis

Theorem 8 (CEP-AOP2 **Integrity**). *Let* CEP-AOP2$[\mathsf{F}, G]$ *be a* CEP-AOP *scheme and a* $MO - nCTXT$ *adversary* $\mathcal{A}$ *making at most* q *queries and* H *is a collision resistant hash function. Then for adversaries* $\mathcal{B}, \mathcal{C}$:

$$\mathsf{Adv}^{\text{mo-nctxt2}}_{\text{CEP-AOP2}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{prg}}_{G}(\mathcal{B}) + \sum_{j=1}^{m \cdot q} \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{F}^{cr}}(\mathcal{C}_j)$$

Theorem 9 (CEP-AOP2 **Confidentiality**). *Let* CEP-AOP = CEP-AOP2$[\mathsf{F}, G]$ *be a* CEP-AOP *scheme,* H *is a collision resistant hash function and a* MO-nRoR *adversary* $\mathcal{A}$ *making at most* $q = q_{\mathsf{Enc}}, q_{\mathsf{Dec}}, q_{\mathsf{Challenge}}$ *queries with time complexity* t. *Then there exist adversaries* $\mathcal{B}$ *making* $q_{\mathsf{PRG}} = q_{\mathsf{Enc}} + q_{\mathsf{Dec}} + q_{\mathsf{Challenge}}$ *and* $\mathcal{C}$ *making* $q_{\mathsf{PRG}} = q_{\mathsf{Enc}} + q_{\mathsf{Dec}} + q_{\mathsf{Challenge}}$ *queries in time complexity* t *each, such that:*

$$\mathsf{Adv}^{\text{mo}-\text{nror}}_{\text{CEP-AOP2}}(\mathcal{A}) \leq 2 \cdot \mathsf{Adv}^{\text{mo-nctxt2}}_{\text{CEP-AOP2}}(\mathcal{B}) + m \cdot \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{F}^{cr}}(\mathcal{C})$$

Theorem 10 (CEP-AOP2 **After Opening Privacy**). *Let* CEP-AOP = CEP-AOP2$[\mathsf{F}, G]$ *be a* CEP-AOP *scheme,* H *is a collision resistant hash function and a* MO-IND-PO *adversary* $\mathcal{A}$ *making* $q = (q_{\mathsf{Enc}}, q_{\mathsf{Dec}}, q_{\mathsf{Proof}}, q_{\mathsf{Challenge}})$ *queries in time complexity* t. *Then there exist adversaries* $\mathcal{B}$ *and* $\mathcal{C}$ *making each* $q_{\mathsf{PRG}} = q_{\mathsf{Enc}} + q_{\mathsf{Dec}} + q_{\mathsf{Challenge}} + q_{\mathsf{Proof}}$ *and* $q_{\mathsf{PRF}} = q_{\mathsf{Enc}} + q_{\mathsf{Dec}} + q_{\mathsf{Challenge}} + q_{\mathsf{Proof}}$ *queries in time complexity* t *such that:*

$$\mathsf{Adv}^{\text{mo-ind-po}}_{\text{CEP-AOP2}}(\mathcal{A}) \leq \mathsf{Adv}^{\text{mo}-\text{nror}}_{\text{CEP-AOP2}}(\mathcal{B}) + m \cdot \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{F}^{cr}}(\mathcal{C})$$

Sender binding is guaranteed as long as decryption algorithm Dec decrypts correctly: it outputs the correct message M or $\perp$ when there is an error, and Verify run by an honest router.

Theorem 11 (CEP-AOP2 **Receiver Binding**). *Let* CEP-AOP2 *be a message franking scheme and* $\mathcal{A}$ *an adversary against* r-BIND *with time complexity* t *and* H *is a collision resistant hash function. Then, there exists an adversary* $\mathcal{B}$ *finding a collision of* F^{cr} *with time complexity* t*:*

$$\mathsf{Adv}^{\text{r}-\text{bind}}_{\text{CEP-AOP2}}(\mathcal{A}) \leq m \cdot \mathsf{Adv}^{\text{cr}}_{\mathsf{F}^{cr}}(\mathcal{B})$$

The proofs of the theorems follow trivially akin to the CEP-AOP1 proofs.

7 Related Work

After Facebook launched their message franking protocol [Fac16] on Facebook Messenger [Fac], Grubbs *et al.* [GLR17] initiated a formal study for verifiable report on abusive messages. The authors first model a new cryptographic primitive called committing AEAD, which demonstrates an encryption scheme with ciphertext integrity, in which part of the ciphertext acts like a commitment for the encrypted plaintext. This flavor of commitment comes with two formal definitions of: a) sender binding, which locks a sender from sending a message that cannot be reported by a receiver and b) receiver binding, which prevents a malicious receiver from reporting successfully a message which has not been sent from a sender. However the notion of achieved privacy excludes the confidentiality of non-abusive blocks. A recent work by Dodis *et al.* [DGRW18] proposes a new committing AEAD scheme with only two passes as in CEP2, but confidentiality is based on the non-standard related-key-attack resistance of the underlying PRF. We stress though that the security model for integrity is the strong one which gives the opening key to an adversary willing to compromise integrity.

A relevant primitive which resembles to receiver binding is the notion of robust PKE [GH03,ABN10,FLPQ13]. Robustness in PKE ensures that a ciphertext cannot be decrypted with two different keys. However, those schemes there are in the public key setting and second they do not model in conjunction with receiver binding the needed sender binding notion for message franking protocols. Farshim *et al.* [FOR17] analyzed Security of Symmetric Primitives under Incorrect Usage of Keys, however that work is not tailored for a message franking protocol as it does not achieve multi-opening security *per se* and the commitment is computed under the key and not the message.

8 Conclusion

In this work we continued the footsteps in message franking protocols for messaging applications. Under this new primitive a receiver of an abusive message can verifiably report it to a third party. We first showed an efficient protocol that captures a weaker but still meaningful notion of MO-nCTXT as with [GLR17]. Namely our CEP2 protocol needs one PRF less, thus reducing the number of passes for encryption and decryption by one evaluation and it lies on standard security assumptions of the underlying symmetric primitives.

We initiated a stronger model for message franking protocols where only the necessary information is opened to the router while the rest is kept privy. We demonstrate the privacy notion with the new MO-IND-PO game definition, for multi-opening indistinguishability with partial openings. In a nutshell, a protocol adhering to MO-IND-PO reveals only the abusive blocks of a message to a third party router, while an adversary cannot break the confidentiality of the remaining blocks, which are not abusive and thus have not been opened to the router. We then model the syntactical framework of a new primitive: Committing Nonce based Authenticated Encryption with Partial Opening and designed

two protocols CEP-AOP1 and CEP-AOP2 (Committing Encrypt and PRF with After Opening Privacy), which are provably secure under the new enhanced security definition. CEP-AOP1 is a vanilla protocol, which treats the blocks of a message as individual information given as input for encryption. This approach increases the computation complexity at the router, as it has to authenticate all individual commitment-tags on blocks separately and the communication complexity for transferring all the commitments-tags from the sender to the router. Our second protocol builds upon Merkle trees as a membership proof for tags of a message. The router now has only to sign the root of the tree and the sender sends only one commitment-tag for the root of the Merkle tree per message.

We leave it as an open problem the design and analysis of message franking protocols with after opening privacy, whereby the communication complexity is reduced to a singleton key instead of the receiver sending all the keys for all the abusive message blocks. Messaging application protocols over the Internet are realized over a secure channel, which apart from confidentiality and integrity of the messages, guarantees resistance to replay attacks and out-of-order delivery. Current literature in secure channels [GM17,MP17a,MP17b,BKN02,BDPS12] is lacking an analysis of a secure channel with message franking properties. Thus, a second future direction is the formal study of a message franking channel with pragmatic properties such as bidirectionality, group messaging and attachment delivery.

References

ABN10. Michel Abdalla, Mihir Bellare, and Gregory Neven. Robust encryption. In *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings*, pages 480–497, 2010.

BDPS12. Alexandra Boldyreva, Jean Paul Degabriele, Kenneth G. Paterson, and Martijn Stam. Security of symmetric encryption in the presence of ciphertext fragmentation. In *Proceedings of the 31st Annual International Conference on Theory and Applications of Cryptographic Techniques*, EUROCRYPT'12, pages 682–699, Berlin, Heidelberg, 2012. Springer-Verlag.

BH05. Boaz Barak and Shai Halevi. A model and architecture for pseudo-random generation with applications to /dev/random. In *Proceedings of the 12th ACM Conference on Computer and Communications Security*, CCS '05, pages 203–212, New York, NY, USA, 2005. ACM.

BKN02. Mihir Bellare, Tadayoshi Kohno, and Chanathip Namprempre. Authenticated encryption in SSH: provably fixing the SSH binary packet protocol. In *Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002, Washington, DC, USA, November 18-22, 2002*, pages 1–11, 2002.

BR00. Mihir Bellare and Phillip Rogaway. Encode-Then-Encipher Encryption: How to Exploit Nonces or Redundancy in Plaintexts for Efficient Cryptography. In Tatsuaki Okamoto, editor, *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, pages 317–330, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

Cal18. Project Callisto. Callisto: Tech to combat sexual assault, 2018. https://www.projectcallisto.org/.

DGRW18. Yevgeniy Dodis, Paul Grubbs, Thomas Ristenpart, and Joanne Woodage. Fast message franking: From invisible salamanders to encryptment. In *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA*, 2018.

DPR+13. Yevgeniy Dodis, David Pointcheval, Sylvain Ruhault, Damien Vergniaud, and Daniel Wichs. Security analysis of pseudo-random number generators with input: /dev/random is not robust. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security*, CCS '13, pages 647–658, New York, NY, USA, 2013. ACM.

Fac. Facebook. Facebook messenger. https://www.messenger.com/.

Fac16. Facebook. Messenger secret conversations technical whitepaper, 2016. https://fbnewsroomus.files.wordpress.com/2016/07/secret_conversations_whitepaper-1.pdf.

FLPQ13. Pooya Farshim, Benoît Libert, Kenneth G. Paterson, and Elizabeth A. Quaglia. Robust encryption, revisited. In *Public-Key Cryptography - PKC 2013 - 16th International Conference on Practice and Theory in Public-Key Cryptography, Nara, Japan, February 26 - March 1, 2013. Proceedings*, pages 352–368, 2013.

FOR17. Pooya Farshim, Claudio Orlandi, and Razvan Rosie. Security of symmetric primitives under incorrect usage of keys. *IACR Trans. Symmetric Cryptol.*, 2017(1):449–473, 2017.

GH03. Yitchak Gertner and Amir Herzberg. Committing encryption and publicly-verifiable signcryption. *IACR Cryptology ePrint Archive*, 2003:254, 2003.

GLR17. Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. Message franking via committing authenticated encryption. In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III*, pages 66–97, 2017.

GM17. Felix Günther and Sogol Mazaheri. A Formal Treatment of Multi-key Channels. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017*, pages 587–618. Springer International Publishing, 2017.

Mer89. Ralph C. Merkle. A certified digital signature. In *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, pages 218–238, 1989.

MP17a. Giorgia Marson and Bertram Poettering. Security notions for bidirectional channels. *IACR Transactions on Symmetric Cryptology*, 2017(1):405–426, 2017.

MP17b. Giorgia Azzurra Marson and Bertram Poettering. With one it is easy, with many it gets complicated: Understanding channel security for groups. *IACR Cryptology ePrint Archive*, 2017:786, 2017.

Sig. Signal. https://signal.org/.

Tel. Telegram. https://telegram.org/.

Vib. Viber. https://www.viber.com/.

Wha. Whatsapp. https://www.whatsapp.com/.