

Application of Deterministic Authenticated Encryption for Storage Area Networks and Block-Oriented Storage Devices Protection

Georgii V. Firsov, Alisa M. Koreneva, Sergey S. Minakov

Abstract—The purpose of this research is to develop a cryptographic scheme CS_{SAN} for block-oriented storage devices and storage area networks protection. This scheme provides data confidentiality and unit-level integrity. By unit-level integrity we mean integrity of each logical data unit (e.g., disk sector) independently. That is, we consider authenticated disk encryption (ADE) notion. The CS_{SAN} scheme could be used as a part of high-level cryptographic scheme, that achieves fully authenticated disk encryption (FADE) notion and controls whole storage integrity even against replay attacks. The proposed scheme is built on top of a deterministic authenticated encryption (DAE) scheme. We use a modified Encrypt-then-MAC (EtM) construction to compose a tweakable encryption scheme and message authentication code scheme into a DAE scheme. We reduce security of the CS_{SAN} scheme to security of its components. Further, we prove security of the modified EtM construction. Based on these security reductions, we formulate requirement for basic cryptographic primitives used in the CS_{SAN} scheme.

Keywords—Block ciphers mode of operation, Block-oriented storage devices, Full disk encryption, Cryptographic protection of information, Provable security, Symmetric cryptography.

I. INTRODUCTION

Storage area network (SAN) is a common architecture of data storage in large enterprise-scale organizations. The SAN is used to make access to remote storage devices (for instance disk arrays) more convenient by mounting it like local drives on computers in the network. In other words, users could use remote storage devices in the exact same ways they use local ones. Typically, SAN provides block-oriented access to storage devices, again in the same way as for local ones.

There exists a variety of technologies used to build storage area networks such as iSCSI (Internet Small Computer System Interface), FCP (Fibre Channel Protocol), SRP (SCSI RDMA Protocol — SCSI Remote Direct Memory Access Protocol), etc. Mentioned protocols form a layer of abstraction that allows to mount remote storage devices like local ones. In Figure 1, we show a subset of

protocol layers involved in remote storage device access system via SAN. The following acronyms are used in Figure 1: SCSI — Small Computer System Interface protocol, NVMe — Non-Volatile Memory Express protocol, PCIe — Peripheral Component Interconnect Express protocol, SMB — Server Message Block protocol, NFS — Network File System protocol, iSER — iSCSI Extension for RDMA, TCP — Transmission Control Protocol, IP — Internet Protocol, FCoE — Fibre Channel over Ethernet, FC — Fibre Channel. From the Figure 1, one could see a variety of technologies and protocols that may be used to build a storage area network.

The data processed in organization (including “data-at-rest” stored in SAN) often contains commercial secrets, personal data, and other types of limited access information. That is, the data processed in organizations should be protected. In [1] authors study security issues of commonly used SANs based on Fibre Channel and iSCSI. The authors claim that the data stored in SAN should be encrypted, but the article does not offer any cryptographic scheme for such purpose. Instead, authors focus on properly configuring SAN, i.e., enabling authentication and authorization.

In [2], authors propose a high-level SAN security framework. Authors show that SAN security consists of large number of interconnected measures functioning together starting from organization’s security policy. To offer cryptographic protection, authors suggest to use encrypted file systems, e.g., CFS (Cryptographic File System), EFS (Encrypted File System), etc. One should not confuse encrypted file systems where each file is encrypted separately, and full disk encryption systems where encryption is performed sector-by-sector. To use encrypted file system, storage device must be properly formatted, whereas full disk encryption systems are transparent for existing file system.

Kim et al. in [3] propose SAN security scheme where all cryptographic operations are performed on SAN’s infrastructure entities. Hosts only need to establish a secure connection to the infrastructure which is assumed to be trusted, because there are entities that observe the data in plain form. However, such configuration does not offer fair end-to-end encryption (with the same user on both “ends” and SAN as a transmission channel) and requires additional resources to be spent on establishing secure connection to the SAN and protecting SAN itself from being compromised. Further, we claim that malicious system administrator could observe and modify the data on infrastructure entities after decryption just before sending it

Manuscript received February 21, 2026.

G. V. Firsov is with National Research Nuclear University MEPhI, Moscow, Russian Federation, Security Code LLC, Moscow, Russian Federation (corresponding author, e-mail: G.Firsov@securitycode.ru).

A. M. Koreneva, PhD is with Security Code LLC, Moscow, Russian Federation (e-mail: A.Koreneva@securitycode.ru).

S. S. Minakov is with Russian Academy of Cryptography, Moscow, Russian Federation (e-mail: ss_minakov@mail.ru).

to a client's host. This scenario imposes additional requirements to adversary models used to assess security. Unfortunately, the article [3] does not contain formal proof of proposed scheme security.

Block Chaining) and CTR-ACPKM (Counter with Advanced Cryptographic Prolongation of Key Material) [13]. In Fig. 2, transmission time of different amounts of data is shown.

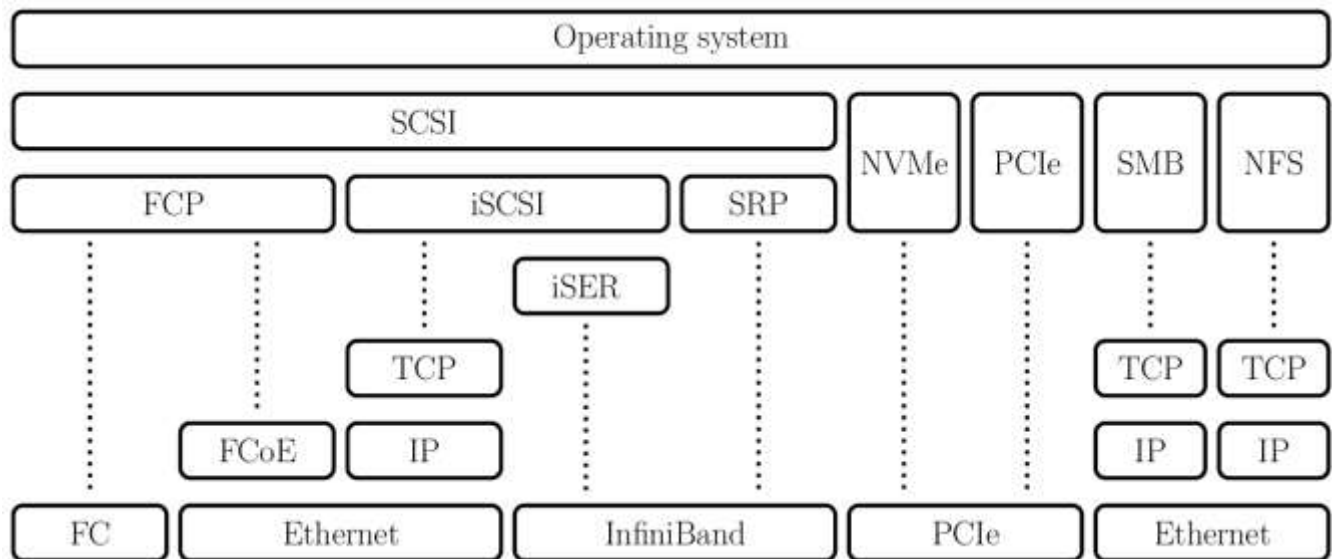


Figure 1. Protocol layers (possible variant) in remote storage devices access systems offered by SAN.

In [4], Wang et al. develop FPGA-based encryption module for storage area networks. The article is focused mainly on performance and does not propose a new encryption scheme. Instead, authors implement a well-known XTS-AES (XTS stands for Xor-Encrypt-Xor based Tweaked codebook with ciphertext Stealing) algorithm. This algorithm has several weaknesses that may lead to theoretical and practical attacks [5, 6]. In fact, the XTS mode of operation is designed to be indistinguishable from tweakable pseudorandom permutation (Tweakable PRP or TPRP) on blocks of length equal to underlying block cipher block size. On the other hand, one could easily distinguish this mode from a tweakable PRP on wide blocks (by wide blocks we mean data segments consisting of two or more underlying block cipher blocks) [7].

In [8], Kim et al. propose a security architecture for preserving data confidentiality and integrity up-to-repetition (however, the authors mention MAC trees that could help mitigating repetition attacks). Authors primarily focus on key management and use Diffie-Hellman-based group key agreement protocol named TGDH (Tree-based Group Diffie-Hellman) and described in [9]. Proposed architecture involves logical parts of a storage device called encrypted data units. Data within an encrypted data unit is protected using same keys, and therefore key exchange operation affects only “relevant” data units instead of whole storage device. Data unit keys are encrypted using a single master key and stored on a storage device, that they belong to. The master key is derived using TGDH protocol and available for all group members (by group member the authors mean a SAN entity). Despite providing a comprehensive description of the proposed security architecture, the article does not contain any formal proof of its security.

In [10] Minakov et al. conduct a series of experiments aimed to estimate average operational overhead caused by encryption of the data stored on remote (cloud) storage device with the following block ciphers models of operation: XEH (Xor-Encrypt-Hash) [11, 12], CBC (Cipher

The plot in Fig. 2 is drawn based on experimental data provided in [10]. The data is transmitted onto the cloud storage via WebDAV protocol. The experiment is conducted on a personal computer with Intel Core i5 13420H 2.1 GHz CPU, 32 GB DDR4 RAM. Each mode of operation is instantiated by Kuznyechik block cipher defined in [14]. Authors claim that measured transmission time includes network channel and WebDAV protocol overhead, and remote storage latency.

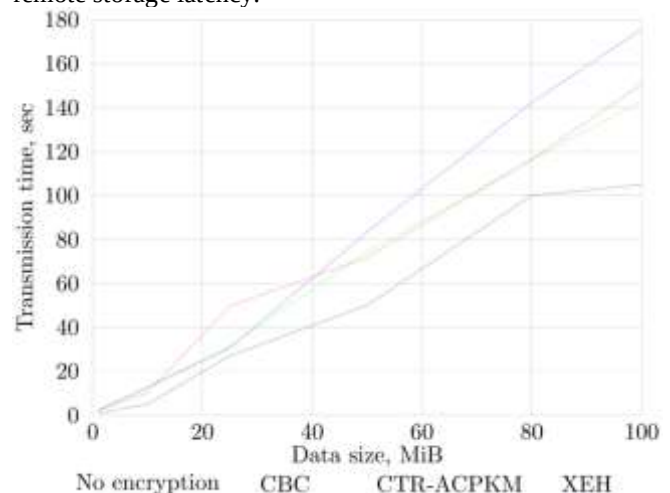


Figure 2. Data transmission time depending on the block ciphers mode of operation used for encryption [10].

In [11], the XEH mode of operation is compared to XTS, CBC (CBC-Mask-CBC) [15], and HEH (Hash-Encrypt-Hash) [16] modes according to their performance. Authors claim that the XEH mode of operation is 35% faster compared to the CMC mode and less than 10% slower than the XTS mode.

Security of remote storages in general (and storage area networks in particular) study was started after a lot of communication protocols had been established. The needs in additional studies of security are induced by remote storage deployment practice due to additional threats. Proposed

solutions (additionally refer to: [17, 18]) rely on securing transmission channels via TLS or IPsec protocols, or encrypting the data on storage infrastructure entities [3]. Such solutions offer protection only against external adversary and assume that infrastructure entities are trusted.

However, storage area networks security model often should take into account absence of the stored data control [19, 20]. That is, the storage provider could not be considered as a trusted party. Therefore, developing a cryptographic scheme for storage area networks is in demand. We emphasize to following difference between cryptographic protocol and cryptographic scheme: the protocol involves at least two participants, whereas the scheme involves only one participant. That is, a cryptographic scheme does not require trusted storage provider. In the present paper we propose a cryptographic scheme for storage area networks and block-oriented storage devices protection.

In [21], Chakraborty et al. apply a deterministic authenticated encryption (this notion appears in [22]) scheme for disk protection. Authors propose a new block cipher mode of operation. However, there are existing algorithms providing only confidentiality or only integrity. In this article we answer the question, if it is possible to combine these algorithms to create a secure scheme for storage area networks or block-oriented storage devices protection. We prove that modified Encrypt-then-MAC construction is secure deterministic authenticated encryption scheme.

The purpose of this paper is to create a cryptographic protection scheme for storage area networks and block-oriented storage devices based on existing algorithms (block cipher modes of operation, message authentication code schemes and key derivation functions). We focus on using of existing algorithms, because many of these algorithms are well-studied now. The goal is to combine them in a way such that composite scheme remains secure, if underlying algorithms are.

The rest of the paper is organized as follows. In Section 2, we introduce main notation and define known security notions. The Section 3 contains definition of the proposed CSSAN scheme. In Section 4, we introduce new security notions and prove the main Theorem about the proposed scheme security. In Section 5, we provide requirements on the scheme components such that the scheme is secure based on reductions of the modified Encrypt-then-MAC construction to security of basic underlying algorithms. The Section 6 consists of the CSSAN scheme instantiation with existing algorithms. In Section 7, we discuss several issues on key lifetime. In Section 8, we summarize results of the current paper.

II. PRELIMINARIES

Basic definitions. We denote the set of all integers (positive integers) by $\mathbb{Z}$ ($\mathbb{N}$). By $\mathbb{N}_0$, we denote the set $\mathbb{N} \cup \{0\}$. The ring of integers modulo l is denoted by $\mathbb{Z}_l$. Sets $1, \dots, l$ and $0, \dots, l$ are denoted by $\mathbb{N}^{\leq l}$ and $\mathbb{N}_0^{\leq l}$ respectively. For the sake of brevity, we define shorthands for $\mathbb{N}^{\leq l-1}$ and $\mathbb{N}_0^{\leq l-1}$, and denote them by $\mathbb{N}^{< l}$ and $\mathbb{N}_0^{< l}$ respectively. The set of positive real numbers is denoted by $\mathbb{R}^+$.

Let $X = \mathbb{Z} \cap [l, m]$ for some $l, m \in \mathbb{Z}$ and $l \leq m$. By writing $j = \overline{l, m}$ after some statement we mean repeating the statement for every $j \in X$.

Let V_l and V^* be the set of all binary strings of length $l \in \mathbb{N}_0$ and the set of all finite binary strings (including empty string λ) respectively.

Concatenation operation of two binary strings is denoted by “||”. Given $a = (a_{l-1}, \dots, a_0) \in V_l$ and $b = (b_{k-1}, \dots, b_0) \in V_k$, the result of a and b concatenation is defined as follows: $a||b = (a_{l-1}, \dots, a_0, b_{k-1}, \dots, b_0) \in V_{l+k}$.

Given a binary string $a = (a_{l-1}, \dots, a_0) \in V_l$, a substring $(a_j, \dots, a_i)$ for $0 \leq i \leq j \leq l-1$ is denoted by $[a]_{i,j}$.

For $l \leq 1$, the function $\text{Bin}_l: \mathbb{Z}_{2^l} \rightarrow V_l$ maps an element $\sum_{j=0}^{l-1} a_j \cdot 2^j$ of the ring $\mathbb{Z}_{2^l}$ to a binary string $(a_{l-1}, \dots, a_0)$. The function $\text{Int}_l: V_l \rightarrow \mathbb{Z}_{2^l}$ is defined as Bin_l^{-1} . Further, we define similar functions for $l = 0$ as follows: $\text{Bin}_0: 0 \mapsto \lambda$ and $\text{Int}_0: \lambda \mapsto 0$.

Given X and Y being arbitrary finite sets, we denote the set of all permutations on X by $S(X)$ and the set of all functions from X to Y by $F(X, Y)$. In this paper, we consider only computable functions. That is, for each function there exists an algorithm (modelled by Turing machine) that computes this function. Later in this paper, whenever we use term “function” we actually mean corresponding algorithm.

The process of choosing and assigning a uniformly random value from a finite set X to a variable x we denote by $x \xleftarrow{\$} X$.

By *random oracle* $\$$ we mean a non-deterministic algorithm that ignores its input (an element of some set X) and returns a uniformly random element of some set Y . The sets X and Y are usually clear from context.

An *encryption scheme* $\mathcal{E}$ is a function $E: \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$, where $\mathcal{M} \neq \emptyset$ is the set of plaintexts, $\mathcal{C} \neq \emptyset$ is the set of ciphertexts, and $\mathcal{K} \neq \emptyset$ is the key space. The key space is finite. For every $K \in \mathcal{K}$ we write $E_K(\cdot)$ for $E(K, \cdot)$. For every $K \in \mathcal{K}$ function $E_K(\cdot)$ is bijective. The function E_K is called *encryption function*, its inverse function is called *decryption function*. *Block cipher* is an encryption scheme, where $\mathcal{M} = \mathcal{C} = V_l$ and $l \in \mathbb{N}$ is a *block size*.

A *tweakable encryption scheme* $\tilde{\mathcal{E}}$ is a function $\tilde{E}: \mathcal{K} \times \mathcal{T}_{tw} \times \mathcal{M} \rightarrow \mathcal{C}$, where $\mathcal{T}_{tw} \neq \emptyset$ is the set of tweaks (tweak is an additional argument of encryption and decryption functions). We write $\tilde{E}_K(\cdot, \cdot)$ to denote $\tilde{E}(K, \cdot, \cdot)$. The function $\tilde{E}_K$ is bijective by its last argument for every $K \in \mathcal{K}$. Corresponding inverse function is denoted by $\tilde{E}_K^{-1}(\cdot, \cdot)$.

An encryption scheme (tweakable or not) is called *length-preserving*, if ciphertext has the same length as plaintext.

In the present paper we are interested in block cipher-based (tweakable) encryption schemes such that $\mathcal{M} = \mathcal{C} = V_{nl}$, where l is a block size of the block cipher, and $n > 1$ is some fixed positive integer. We emphasize, that we focus on length-preserving schemes. Such schemes are usually built using specially designed block cipher modes of operation¹. We call these modes *wide block encryption* (WBE) modes.

¹ There is another approach for building encryption schemes for wide blocks without using a “conventional” block ciphers. The key idea is to build a block cipher that operates on wide blocks (wider than

A *message authentication code scheme* MAC is a pair of functions (S, V) , where $S: \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{T}_{tag}$, $V: \mathcal{K} \times \mathcal{M} \times \mathcal{T}_{tag} \rightarrow \{0, 1\}$, $\mathcal{K}$ is the key space, $\mathcal{M}$ is the set of messages, $\mathcal{T}_{tag}$ is the tag space. These functions are such that $V(K, m, S(K, m)) = 1$. The functions S and V are called *signing function* and *tag verification function* respectively.

A *key derivation function* KDF is a function $F: \mathcal{K} \times V^* \times \mathbb{N} \rightarrow V^*$. The function takes a primary key $K \in \mathcal{K}$, a binary string $x \in V^*$ of finite length and a number $k \in \mathbb{N}$, and returns a binary string of length k . We use the following notation: F denotes key derivation function in mathematical statements, KDF denotes key derivation function as a cryptographic primitive.

Deterministic authenticated encryption. The definition of deterministic authenticated encryption (DAE) is adapted from [22].

A *deterministic authenticated encryption scheme* $\tilde{\mathcal{E}}^2$ is a pair of functions $(\tilde{E}, \tilde{E}^{-1})$, where $\tilde{E}: \mathcal{K} \times \mathcal{T}_{tw} \times \mathcal{M} \rightarrow \mathcal{C} \times \mathcal{T}_{tag}$ and $\tilde{E}^{-1}: \mathcal{K} \times \mathcal{T}_{tw} \times \mathcal{C} \times \mathcal{T}_{tag} \rightarrow \mathcal{M} \cup \{\perp\}$, $\mathcal{T}_{tw}$ is a tweak space, and $\mathcal{T}_{tag}$ is a tag space. The encryption function $\tilde{E}$ produces a ciphertext and tag, the decryption function $\tilde{E}^{-1}$ verifies tag, produces plaintext in case of the tag being valid and the special value $\perp$ otherwise.

Our definition differs in several points from original definition in [22] given by Rogaway and Shrimpton. First, we replace *header space* $\mathcal{H}$ by the tweak space $\mathcal{T}_{tw}$. We use term tweak space to emphasize similar ideas behind tweakable encryption schemes and deterministic authenticated encryption schemes. Next, in original definition encryption and decryption functions return binary strings or a special value $\perp$. We define encryption (decryption) function to return ciphertext and tag (plaintext or special value $\perp$). However, these differences are mostly notational.

Definition of the deterministic authenticated encryption scheme is similar to some other definitions, but differs from them in several points. Conventional nonce-based authenticated encryption scheme (NAE) requires unique nonce to be used on each encryption [24], the DAE does not. Robust authenticated encryption (RAE) requires nonce and provides a way to specify ciphertext extension size [25], the DAE does not. Nonce-misuse resistant authenticated encryption (MRAE) [22] does not guarantee confidentiality of data processed with repeated nonce, the DAE does not require a nonce by definition.

The DAE scheme is suitable for disk (or storage device in general) encryption purposes in following sense. The scheme is length-preserving, and the only additional data appearing after encryption is a tag³. Therefore, such scheme could be implemented via one-to-one correspondence (i.e. without complex remapping of logical and physical disk

layout) of original and encrypted data units (smallest addressable chunks of data, see Subsection 2.4) with writing tags in a special relatively small area in storage device. The smaller additional data is, the more user data could be protected. In this sense, the DAE scheme is optimal, because the only additional data are tags.

One security model for DAE is present in the Section 4. This security model differs from originally defined in [22] to conform disk encryption limitations and restrictions.

Provable security. By *provable security model* (hereinafter we use terms *model* or *security model*) we mean a probabilistic experiment, where a probabilistic algorithm called adversary $\mathcal{A}$ interacts with a set of oracles $\mathcal{O}_1, \dots, \mathcal{O}_w$ for $w \in \mathbb{N}$. These oracles (at least in one form of the experiment) implement some algorithms of an analyzed cryptographic system CS.

The adversary sends several queries to the oracles and receives responses. If the adversary fixes all its queries a priori (i.e. before any interaction with oracles), the model is called *non-interactive*, otherwise the model is called *interactive*. Hereinafter, we consider only interactive models.

After interaction with the oracles, the adversary $\mathcal{A}$ returns some result x . We denote this fact by writing $\mathcal{A}^{(\mathcal{O}_1, \dots, \mathcal{O}_w)} \Rightarrow x$.

Given a security model MODEL-ATK, we denote quantitative characteristic of the adversary $\mathcal{A}$ possibilities (we use term *advantage* for this characteristic) by $\text{Adv}_{\text{CS}}^{\text{MODEL-ATK}}(\mathcal{A})$.

Let $\{a_1, \dots, a_u\}$, $u \in \mathbb{N}_0$ be a set of the adversary $\mathcal{A}$ resources (number of queries q , time or number of elementary operations t , etc.) restrictions. By $\mathcal{A}(a_1, \dots, a_u)$ we denote the set of all adversaries such that their resources are upper bounded by $\{a_1, \dots, a_u\}$. We denote the maximum advantage among all adversaries in $\mathcal{A}(a_1, \dots, a_u)$ by:

$$\begin{aligned} \text{Adv}_{\text{CS}}^{\text{MODEL-ATK}}(a_1, \dots, a_u) \\ = \max_{\mathcal{A} \in \mathcal{A}(a_1, \dots, a_u)} \text{Adv}_{\text{CS}}^{\text{MODEL-ATK}}(\mathcal{A}). \end{aligned}$$

Cryptosystem CS is called secure in MODEL-ATK model, if there exists a value ϵ (more formally ϵ is a function of security parameter ξ^4 that formalizes security requirements) such that:

$$\text{Adv}_{\text{CS}}^{\text{MODEL-ATK}}(a_1, \dots, a_u) \leq \epsilon,$$

where the set of restrictions $\{a_1, \dots, a_u\}$ matches practical considerations.

From definition of the $\mathcal{A}(a_1, \dots, a_u)$ set, one could derive the following statement: if restriction sets $\{a_1, \dots, a_u\}$ and $\{b_1, \dots, b_u\}$ are such that $a_j \leq b_j$, $j = \overline{1, u}$, then $\mathcal{A}(a_1, \dots, a_u) \subseteq \mathcal{A}(b_1, \dots, b_u)$.

One of widely applied model classes is *distinguishing models* class. We define distinguishing model in general form. The goal of the adversary is to distinguish between two cryptosystems CS_0 and CS_1 . Before the experiment a binary value $b \in \{0, 1\}$ is uniformly chosen. The oracles $\mathcal{O}_1^b, \dots, \mathcal{O}_w^b$ implement functions of the CS_b cryptosystem. After interaction with oracles, the adversary $\mathcal{A}$ returns a

² “conventional” block cipher operates on). Refer to [23] for a example of such construction. However, we do not consider this approach in the present paper.

³ By abuse of notation, we denote tweakable encryption scheme and deterministic authenticated encryption scheme by the same symbol.

⁴ In original definition from [22], Rogaway and Shrimpton define a notion of expansion function, and call the scheme length-preserving if and only if the expansion function is constant zero. However, they consider ciphertext to include tag, we do not.

⁴ Usually, the security parameter is denoted by λ , but in this paper we use this symbol to denote the empty binary string.

binary value b' (a guess for the unknown value b). Specific restrictions on queries are defined by concrete model. The adversary $\mathcal{A}$ advantage (also referred to as *distinguishing advantage*) is defined as follows:

$$\text{Adv}_{\text{CS}_0, \text{CS}_1}^{\text{IND}}(\mathcal{A}) = \Pr[\mathcal{A}(\mathcal{O}_1^0, \dots, \mathcal{O}_w^0) \Rightarrow 1] - \Pr[\mathcal{A}(\mathcal{O}_1^1, \dots, \mathcal{O}_w^1) \Rightarrow 1]. \quad (1)$$

We also use the following notation, where the way of building oracles $\mathcal{O}_1^b, \dots, \mathcal{O}_w^b$ on top of CS_b is clear from context:

$$\text{Adv}_{\text{CS}_0, \text{CS}_1}^{\text{IND}}(\mathcal{A}) = \Pr[\mathcal{A}^{\text{CS}_0} \Rightarrow 1] - \Pr[\mathcal{A}^{\text{CS}_1} \Rightarrow 1].$$

Some authors define distinguishing advantage to be an absolute value of difference (1) to avoid negative advantage values. However, without loss of generality we could focus only on adversaries such that difference (1) is non-negative. Indeed, for every adversary $\mathcal{A}$ such that $\text{Adv}_{\text{CS}_0, \text{CS}_1}^{\text{IND}}(\mathcal{A}) = x < 0$, we could build an adversary $\mathcal{B}$ such that $\text{Adv}_{\text{CS}_0, \text{CS}_1}^{\text{IND}}(\mathcal{B}) = -x$ simply by inverting result of $\mathcal{A}$ ⁵. The latter equality directly follows from (1).

The *PRP model* (pseudorandom permutation) is a distinguishing model, where an adversary $\mathcal{A}$ is trying to distinguish a function $F: \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{X}$ and a truly random permutation on $\mathcal{X}$ ($\mathcal{K}$ and $\mathcal{X}$ are both finite sets). That is, the adversary interacts with a single oracle by sending queries consisting of a value $x \in \mathcal{X}$. In the $b = 0$ case, the oracle responds with value of $F(K, x) = F_K(x)$, where $K \in \mathcal{K}$ is uniformly chosen before accepting queries. In the $b = 1$ case, the oracle responds with $\pi(x)$, where π is uniformly chosen from the set $S(\mathcal{X})$ before handling queries. The advantage of the adversary $\mathcal{A}$ is defined in a straightforward way:

$$\text{Adv}_F^{\text{PRP}}(\mathcal{A}) = \text{Adv}_{F_K, \pi}^{\text{IND}}(\mathcal{A}).$$

The $\pm$ PRP model adds a second oracle to the PRP model. This oracle responds with $F_K^{-1}(x)$ or $\pi^{-1}(x)$ to the query x depending on initial choice of b .

The *PRF model* (pseudorandom function) is similar to the PRP model. The goal of an adversary $\mathcal{A}$ is to distinguish a $F: \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{Y}$ and a truly random function R chosen from $F(\mathcal{X}, \mathcal{Y})$ ($\mathcal{K}$, $\mathcal{X}$, $\mathcal{Y}$ are finite sets). That is, the difference between the PRF and the PRP models is that in the latter one the function F is not necessary a permutation. The advantage of the adversary $\mathcal{A}$ is defined as follows:

$$\text{Adv}_F^{\text{PRF}}(\mathcal{A}) = \text{Adv}_{F_K, R}^{\text{IND}}(\mathcal{A}).$$

The *VO-PRF model* (variable output pseudorandom function) is an extension of the PRF model. Consider a function $F: \mathcal{K} \times V^* \times \mathbb{N} \rightarrow V^*$ ⁶. The difference from the PRF model is that the function has additional argument that represents the output length. Hence, each query in the VO-PRF model is of form $(x, n) \in V^* \times \mathbb{N}$. Advantage of the adversary $\mathcal{A}$ is defined as a distinguishing advantage between $F_K(\cdot) = F(K, \cdot)$ and a truly random function $R \in F(V^* \times \mathbb{N}, V^*)$:

$$\text{Adv}_F^{\text{VO-PRF}}(\mathcal{A}) = \text{Adv}_{F_K, R}^{\text{IND}}(\mathcal{A}).$$

⁵ Informally, if some adversary rarely guesses correctly, another adversary could simply invert the result of the former one, and therefore rarely output incorrect guesses

⁶ In the VO-PRF model, we fix input and output sets to be V^* , however the model could be properly defined for arbitrary sets with the positive-valued function $|\cdot|$ defined on the output set.

We briefly describe another distinguishing model — the *RND-fdeCCA-sector model* defined in [12]. Let $\tilde{\mathcal{E}}$ be a tweakable encryption scheme. An adversary $\mathcal{A}$ interacts with encryption and decryption oracles. Encryption and decryption queries are of form $(T, m) \in \mathcal{T}_{tw} \times \mathcal{M}$ and $(T, c) \in \mathcal{T}_{tw} \times \mathcal{C}$ respectively. We assume that $\mathcal{M} = \mathcal{C} = V_{nl}$ for $n, l \in \mathbb{N}^7$. In the $b = 0$ case, the oracles respond with ciphertext and plaintext respectively obtained via the $\tilde{\mathcal{E}}$ scheme. In the $b = 1$ case, the oracles respond with values uniformly and independently chosen from the V_{nl} set (we denote such oracle by $\$,$ i.e. it is a random oracle). The adversary never repeats its queries, never queries (T, m) if m has been received from the second oracle as response on query with the same T , and vice versa. The advantage is defined as follows:

$$\begin{aligned} \text{Adv}_{\tilde{\mathcal{E}}}^{\text{RND-fdeCCA-sector}}(\mathcal{A}) \\ = \text{Adv}_{(\tilde{E}_K, \tilde{E}_K^{-1}), (\$, \$)}^{\text{IND}}(\mathcal{A}). \end{aligned}$$

The RND-fdeCPA-sector model is defined in similar way as the RND-fdeCCA-sector is, except for the adversary being restricted to perform encryption queries only.

Definition of the *SUF-CMA model* (Strong UnForgeability under Chosen Message Attack) is adapted from [26]. Let MAC be a message authentication code scheme implemented by two oracles. The first oracle accepts a message and responds with its tag. The second oracle accepts a message-tag pair and responds with a tag verification result. An adversary $\mathcal{A}$ performs several queries to the first oracle and then exposes a new message-tag pair (forgery attempt) to the second oracle. By new we mean that exposed pair has never appeared during queries to the first oracle. Advantage of $\mathcal{A}$ is defined as successful (i.e. the second oracle returns 1) forgery probability:

$$\text{Adv}_{\text{MAC}}^{\text{SUF-CMA}}(\mathcal{A}) = \Pr[\mathcal{A}^{\text{MAC}} \text{ forges}].$$

Proofs of composite schemes security are based on reductions of the composite scheme security to security of its building blocks (components). In fact, these proofs utilize one basic principle of formal logic known as modus tollens [27]:

$$\frac{P \Rightarrow Q, \neg Q}{\therefore \neg P},$$

where the statement P is “components are secure”, and the statement Q is “composite system is secure”. In general, reduction is built as follows: we suppose that all components are secure (in some models), and the composite system is insecure (in some model), then we construct an attack on each component. This leads to obvious contradiction with initial assumptions and completes the proof.

Data storage model. Data storage (storage area network or block-oriented storage device) consists of several storage objects that have unique numbers denoted by $ON \in \mathbb{N}_0$. For iSCSI-based SAN's ON is exactly a logical unit number (LUN), for hard disk drives (HDD) or solid-state drives (SSD) ON is a partition number. Each storage object consists of data units which are the smallest addressable chunks of consecutive data. Each data unit has its own number $UN \in \mathbb{Z}_2^l$ unique within a storage object. For

⁷ The crucial part of this assumption is that message and ciphertext spaces are both the set of binary string of some fixed length. However, it would be convenient to assume these strings to consist of n l -bit blocks.

iSCSI-based SAN's UN is logical block address (LBA), for HDD's and SSD's UN is a sector number. We model data storage with $N \in \mathbb{N}$ storage objects as an N-tuple $\mathcal{DS} = (\mathcal{S}_0, \dots, \mathcal{S}_{N-1})$. Hereinafter we assume the storage device to have N storage objects. For every $ON \in \mathbb{N}_0^{<N}$ the set $\mathcal{S}_{ON}$ is a subset of $V_{nl} \times V_m \times \mathbb{Z}_{2^l}$, where l is the blocksize of a block cipher used to protect the storage, n is the number of blocks in data unit (hereinafter we assume the data unit length in bits being a multiple of blocksize), m is the tag size in bits. That is, each storage object is a set of triples, where first element is a ciphertext, second one is a tag, and the last one is a unit number. There could not be two triples in the same storage object with the same last element.

III. PROPOSED SAN PROTECTION SCHEME

In this section, we define the CS_{SAN} scheme of cryptographic information protection for storage area networks and block-oriented storage devices. First, we focus on several assumptions. The scheme involves two parties: owner of information $\mathcal{O}$ (for the sake of brevity we would use the term "owner") and data storage $\mathcal{S}$. We emphasize, that only the owner $\mathcal{O}$ performs all cryptographic operations, and hence, CS_{SAN} still could be named a cryptographic scheme (not a protocol). The second participant (the storage $\mathcal{S}$) is introduced only for convenience and could be safely eliminated from the scheme definition. We assume, that all interaction protocols, storage objects and data unit's numbers, and all required data related to interactions are known to both participants a priori.

More precisely, the CS_{SAN} scheme is denoted by $\text{CS}_{\text{SAN}}[\mathcal{E}, \text{MAC}, \text{KDF}, p]$, where $\mathcal{E} = \text{WBE}[\mathcal{E}]$ is a tweakable encryption scheme based on a block cipher $\mathcal{E}$ and a wide block encryption mode WBE^8 , MAC is message authentication code scheme, KDF is a key derivation function, and $p \in \mathbb{Z}_l$ is a *partitioning threshold* parameter which we explain later in this section.

We define the sets involved in definitions of $\mathcal{E}$, MAC, and KDF as follows. We assume that the messages and ciphertexts sets of $\mathcal{E}$ ($\mathcal{M}_e$ and $\mathcal{C}_e$ respectively⁹) are exactly V_{nl} , where l is a blocksize of underlying block cipher, n is the number of blocks in data unit. The tweaks set $\mathcal{T}_{tw}$ is V_l . The key space $\mathcal{K}_e$ is V_{k_e} for $k_e \in \mathbb{N}^{10}$.

The messages set of MAC (denoted by $\mathcal{M}_m^{11}$) is $\mathcal{C}_e = V_{nl}$. The key space $\mathcal{K}_m$ of MAC is V_{k_m} for $k_m \in \mathbb{N}$. The set of tags $\mathcal{T}_{tag}$ is V_{t_m} for $t_m \in \mathbb{N}$.

Each storage object within a data storage has two independent master keys: K_e^{ON} for encryption and K_m^{ON} for message authentication. The storage object has two sets associated with it. First one contains encryption keys, we denote this set by $\mathcal{K}_e^{ON}$. The second one consists of message authentication keys and is denoted by $\mathcal{K}_m^{ON}$. We describe how these sets are initialized later in this section.

Consider p most significant bits of unit number $UN_{msb} = [\text{Bin}_l(UN)]_{l-p-1, l-1}$. Units with numbers UN having the same UN_{msb} values are protected using the same keys derived from corresponding object's master keys and UN_{msb} value.

The CS_{SAN} scheme is fully described by three algorithms called $\text{CS}_{\text{SAN}}.\text{KeyGen}$, $\text{CS}_{\text{SAN}}.\text{Read}$, and $\text{CS}_{\text{SAN}}.\text{Write}$.

Before defining these algorithms, we introduce helper algorithms $\text{DKey}_k^{F,p}$, ReadData , WriteData , and GetKeys^p as shown in Fig. 3, 4, 5, and 6 respectively.

Algorithm $\text{DKey}_k^{F,p}(K, j)$
 Invoke key derivation function
 return $F(K, \text{Bin}_p(j), k)$

Figure 3. The $\text{DKey}_k^{F,p}$ helper algorithm.

Algorithm $\text{ReadData}(ON, UN)$
 Check object number
 if $ON > N$ then
 return $\perp$
 end
 Check unit number and find data
 if $\exists c, \text{tag}$ s.t. $(c, \text{tag}, UN) \in \mathcal{S}_{ON}$ then
 return c, tag
 end
 return $\perp$

Figure 4. The ReadData helper algorithm.

Algorithm $\text{WriteData}(c, \text{tag}, ON, UN)$
 Check object number
 if $ON > N$ then
 return $\perp$
 end
 Write data (with replacement)
 if $\exists c', \text{tag}'$ s.t. $(c', \text{tag}', UN) \in \mathcal{S}_{ON}$ then
 $\mathcal{S}_{ON} \leftarrow \mathcal{S}_{ON} \setminus (c', \text{tag}', UN)$
 end
 $\mathcal{S}_{ON} \leftarrow \mathcal{S}_{ON} \cup (c, \text{tag}, UN)$

Figure 5. The WriteData helper algorithm.

Algorithm $\text{GetKeys}^p(\mathcal{K}_e^{ON}, \mathcal{K}_m^{ON}, UN)$
 Get p most significant bits of UN
 $UN_{msb} \leftarrow [\text{Bin}_l(UN)]_{l-p-1, l-1}$
 $j \leftarrow \text{Int}_p(UN_{msb})$
 Get corresponding keys
 $K_e \leftarrow K_j$ s.t. $(K_j, j) \in \mathcal{K}_e^{ON}$
 $K_m \leftarrow K_j$ s.t. $(K_j, j) \in \mathcal{K}_m^{ON}$
 return K_e, K_m

Figure 6. The GetKeys^p helper algorithm.

The $\text{CS}_{\text{SAN}}.\text{KeyGen}$ algorithm establishes keys used to encrypt/decrypt data and control its integrity. This algorithm is shown in Fig. 7. It takes two master keys K_e^{ON} and K_m^{ON} that correspond to the storage object with number ON , and returns two sets of keys for encryption and message authentication ($\mathcal{K}_e^{ON}$ and $\mathcal{K}_m^{ON}$ respectively). If partitioning threshold p equals 0, then master keys are the only elements of these sets. Otherwise, the sets consist of derived via key derivation function keys for each number $j \in \mathbb{N}_0^{<2^p}$. The

⁸ By WBE we denote an arbitrary (but fixed) wide block encryption mode from a class, e.g. the XEH mode [12].

⁹ Subscript "e" stands for "encryption".

¹⁰ Assumptions about the sets come from practice, i.e. most of widely used block cipher-based encryption schemes operate on binary strings.

¹¹ Subscript "m" stands for "MAC".

CS_{SAN} .KeyGen algorithm is invoked only on the owner's $\mathcal{O}$ side and does not involve any I/O operations.

Algorithm CS_{SAN} .KeyGen(K_e^{ON}, K_m^{ON})

```

if  $p = 0$  then
   $K_e^{ON} \leftarrow \{K_e^{ON}\}$ 
   $K_m^{ON} \leftarrow \{K_m^{ON}\}$ 
else
   $K_e^{ON} \leftarrow \left\{ (K_j, j) \mid \begin{array}{l} K_j \leftarrow DKey_{K_e}^{F,p}(K_e^{ON}, j) \\ j = \overline{0, 2^p - 1} \end{array} \right\}$ 
   $K_m^{ON} \leftarrow \left\{ (K_j, j) \mid \begin{array}{l} K_j \leftarrow DKey_{K_m}^{F,p}(K_m^{ON}, j) \\ j = \overline{0, 2^p - 1} \end{array} \right\}$ 
end
return  $K_e^{ON}, K_m^{ON}$ 

```

Figure 7. The CS_{SAN} .KeyGen algorithm.

The CS_{SAN} .Read algorithm involves both owner $\mathcal{O}$ and data storage $\mathcal{S}$ parties. Given ON and UN on the owner's side, the owner $\mathcal{O}$ sends these numbers to the data storage $\mathcal{S}$. Data storage retrieves the data from unit with number

Figure 8. The CS_{SAN} .Read algorithm.

Algorithm CS_{SAN} .Read(ON, UN)

Owner $\mathcal{O}$ K_e^{ON}, K_m^{ON} ON, UN	Data storage $\mathcal{S}$
	$\xrightarrow{ON, UN}$
	$\xleftarrow{d} \quad d \leftarrow \text{ReadData}(ON, UN)$
<pre> if $d = \perp$ then return $\perp$ end $K_e, K_m \leftarrow \text{GetKeys}^p(K_e^{ON}, K_m^{ON}, UN)$ $(c, tag) \leftarrow d$ if $V(K_m, c UN, tag) = 0$ then return $\perp$ end return $\tilde{E}_{K_e}^{-1}(UN, c)$ </pre>	

UN of storage object with number ON . The data consists of ciphertext c and corresponding message authentication tag tag or a special value $\perp$ that denotes reading error (e.g. incorrect ON or UN). The data storage sends retrieved data back to the owner. The owner checks if the data storage is not $\perp$, then verifies the tag, and if it is correct, decrypts the ciphertext using UN as a tweak. The CS_{SAN} .Read algorithm is shown in Fig. 8.

The CS_{SAN} .Write algorithm requires both parties to participate. Given ON , UN , and plaintext m , the owner $\mathcal{O}$ encrypts the plaintext and computes MAC for obtained ciphertext. Then the owner sends the ciphertext, corresponding tag, ON , and UN to the data storage $\mathcal{S}$. The data storage writes the data to unit with number UN of storage object with number ON . The CS_{SAN} .Write algorithm is shown in Fig. 9.

From definition of the CS_{SAN} .Read and CS_{SAN} .Write algorithms, one could note, that all cryptographic operations

are performed on the owner $\mathcal{O}$ side. The data storage $\mathcal{S}$ does not know secret keys and therefore does not observe the data in the plain form.

To access the storage $\mathcal{S}$ from multiple devices (e.g. computers in an organization), the owner $\mathcal{O}$ should share master keys between these devices. Methods of key distribution are out of scope of the present research.

IV. SECURITY ANALYSIS

In this section we study security properties of the CS_{SAN} scheme. Given the $CS_{SAN}[\mathcal{E}, \text{MAC}, \text{KDF}, p]$ scheme, we denote a corresponding deterministic authenticated encryption scheme by $AE_{SAN}[\mathcal{E}, \text{MAC}]$ (if parameters are clear from context, we denote this scheme simply by AE_{SAN}). We consider cases $p = 0$ and $p > 0$ separately.

Informally, we consider that an adversary knows everything about cryptographic scheme in use except for key material. That is, the adversary does not know master keys and derived ones. The adversary could:

- observe all the data transmitted from the data owner $\mathcal{O}$ and the storage $\mathcal{S}$ (i.e. observe ciphertexts and authentication tags);
- intercept, modify and send further any data packet it observes (i.e. modify ciphertexts and authentication tags);
- remember any stored information and write it into any location in the storage $\mathcal{S}$ at any moment of time (read and replace encrypted data with previously observed ciphertext and authentication tags).

The adversary's goals are:

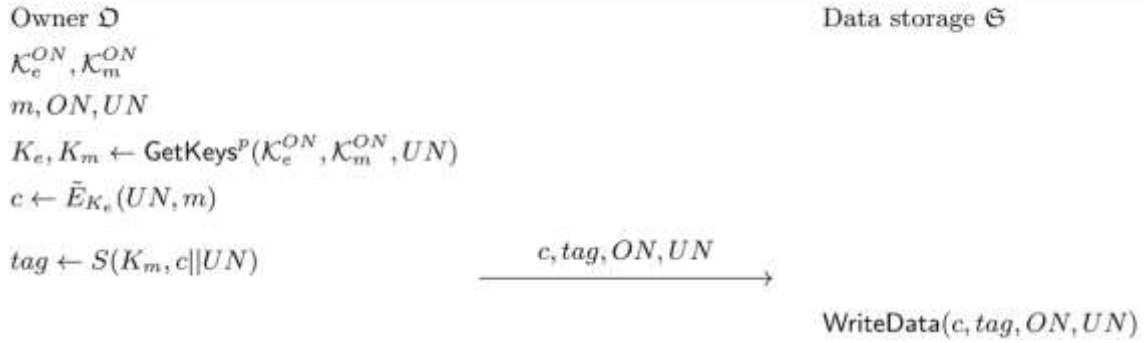
- obtain any information about plaintexts by observing ciphertexts (except for plaintext length in bits);
- forge any message that has not been previously observed by the adversary;

- forge a message that has been previously observed by the adversary by writing it instead of current data stored in SAN.

However, the last goal could be easily achieved by the adversary in the following way: suppose the adversary reads the data (and corresponding authentication tag) located in unit with number UN on storage object with number ON , then after some time being spent, the adversary replaces the data (and corresponding authentication tag) in the same unit

Figure 9. The $CS_{SAN}.Write$ algorithm.

Algorithm $CS_{SAN}.Write(m, ON, UN)$



on the same storage object with previously read one. In this case, the data owner $\mathcal{O}$ is unable to detect forgery without additional security measures. For instance, one could use incremental message authentication code schemes for whole storage integrity protection [28] together with writing the storage tag into a protected memory. In this article, we consider the following adversary goal instead of the last one in the list above: the adversary's goal is to forge a message that has been previously observed by writing it instead of current data stored in SAN *in another location (e.g. in another unit on the same storage object, or in any unit on another storage object)*.

Recall a “weaker” adversary goal: to distinguish the CS_{SAN} scheme from an ideal scheme. Indeed, the adversary that achieves goals mentioned above could be easily converted into a distinguisher that achieves the distinguishing goal. For instance, if the adversary could put a previously observed message in another location and this forgery would not be detected, then one could construct a distinguisher as follows. The adversary performs an encryption query, observes a ciphertext and corresponding tag, then it performs a decryption query (forgery attempt). Real scheme returns a valid plaintext, whereas the ideal scheme rejects this decryption query.

That is, if we formally prove that no adversary could distinguish the CS_{SAN} scheme from an ideal one given some operational restrictions, then we could claim that given the same restrictions no adversary could achieve the goals mentioned in this section above.

Security models. Now we define a security model for deterministic authenticated encryption. We call this model *RND-adeCCA-sector* (indistinguishability from random strings under chosen ciphertext attack with authenticated disk encryption restrictions). This model is similar to IND-

CCA3 [29], NAE [24, 30], and DAE [22] models, but differs in some restrictions on the adversary queries and cryptosystem type being analyzed. Authenticated disk encryption (ADE) notion extends full disk encryption (FDE) notion by adding an authentication tag to each data unit of storage device [31]. There is a stronger notion — FADE (Fully authenticated disk encryption) which ensures integrity of a whole storage (even against replay attacks) [31]. In the present research, we focus on the ADE notion

and leave the FADE notion beyond the scope.

Let $\tilde{\mathcal{E}}$ be a DAE scheme. For simplicity, we assume, that $\mathcal{M} = \mathcal{C} \subseteq V^{*12}$.

In the RND-adeCCA-sector model, the adversary $\mathcal{A}$ interacts with a pair of oracles that correspond to encryption and decryption functions of the analyzed scheme. The RND-adeCCA-sector model is a probabilistic experiment that has two forms. By convention, we use terms “real” and “random worlds” to address these forms of the experiment. In the “real world”, the adversary $\mathcal{A}$ interacts with the $\tilde{\mathcal{E}}$ scheme. In the “random world”, the adversary interacts with pair $(\$, \perp)$ of ideal oracles.

The adversary performs q_e queries of form (T, m) , where $T \in \mathcal{T}_{tw}$ and $m \in \mathcal{M}$ to the “left” oracle (encryption function $\tilde{E}$ of the $\tilde{\mathcal{E}}$ scheme or $\$$), and q_m queries of form (T, c, tag) , where $c \in \mathcal{C}$, $tag \in \mathcal{T}_{tag}$ to the “right” oracle (decryption function $\tilde{E}^{-1}$ of the $\tilde{\mathcal{E}}$ scheme or $\perp$). The suffix “sector” in the model name means that all plaintexts and ciphertexts are of the same length.

In the “real world”, the “left” oracle returns (c, tag) , where $c \in \mathcal{C}$ and $tag \in \mathcal{T}_{tag}$. In the “random world”, the “left” oracle returns two uniformly random independent binary strings of the same lengths as returned by “left” oracle in the “real world”. The “right” oracle in the “real world” returns $m \in \mathcal{M} \cup \{\perp\}$. In the “random world”, the “right” oracle always returns the special symbol $\perp$.

The adversary never repeats its queries, never queries (T, tag, c) to the “right” oracle if (tag, c) was previously received from the “left” oracle and vice versa.

¹² The RND-adeCCA-sector model could be defined in a more general setting assuming $\mathcal{M}$ and $\mathcal{C}$ being arbitrary equal sets. However, the assumption of them being sets of binary strings conforms practical considerations.

The adversary $\mathcal{A}$ advantage is defined as distinguishing advantage between $\tilde{\mathcal{E}}$ and $(\$, \perp)$:

$$\text{Adv}_{\tilde{\mathcal{E}}}^{\text{RND-adeCCA-sector}}(\mathcal{A}) = \text{Adv}_{(\tilde{\mathcal{E}}, \tilde{\mathcal{E}}^{-1}), (\$, \perp)}^{\text{IND}}(\mathcal{A}).$$

Next, we define two helper models that are used to determine requirements on the CS_{SAN} scheme components. The *INT-adeCTXT-sector model* (integrity of ciphertexts with authenticated disk encryption restrictions) formalizes security of a DAE scheme $\tilde{\mathcal{E}}$ against adversaries aimed to modify ciphertexts. In this model an adversary $\mathcal{A}$ interacts with encryption and decryption oracles. Encryption queries are of form (T, m) , where $T \in \mathcal{T}_{\text{tw}}$, $m \in \mathcal{M}$. The encryption oracle responds with (c, tag) , where $c \in \mathcal{C}$, $\text{tag} \in \mathcal{T}_{\text{tag}}$. Decryption queries are of form (T, c, tag) . Decryption oracle responds with a plaintext m , if the tag is valid, or special value $\perp$ otherwise. The suffix “sector” in the model name means that all plaintexts and ciphertexts are of the same length. The adversary goal is to forge a ciphertext, i.e. to send a query to the decryption oracle such that the latter returns a valid plaintext. By valid forgery attempt we mean a decryption query (T, c, tag) such that (c, tag) is not obtained from encryption oracle in response to (T, m) . If different tweaks are used in these queries, then forgery attempt is valid. Only valid forgery attempts are allowed.

The adversary advantage is defined as probability of successful forgery (at any decryption query):

$$\text{Adv}_{\tilde{\mathcal{E}}}^{\text{INT-adeCTXT-sector}}(\mathcal{A}) = \Pr[\mathcal{A}^{(\tilde{\mathcal{E}}, \tilde{\mathcal{E}}^{-1})} \text{ forges}].$$

The *INT-adeCTXT1-sector model* is defined in the same way as the *INT-adeCTXT-sector model*, but restricts decryption queries to a single one. The advantage of an adversary $\mathcal{A}$ is defined exactly as for the *INT-adeCTXT-sector model*.

The *INT-adeCTXT-sector* and *INT-adeCTXT1-sector* models are similar to the *INT-CTXT* [26], *auth*, and *auth1* [32] models.

Case $p = 0$. In our proofs, we often build an adversary on top of another one. The former and the latter are called *wrapper adversary* and *wrapped adversary* respectively. While estimating time wrapper adversary performs in, we consider overhead associated with queries and responses forwarding (from the wrapped adversary to the wrapper’s oracles and vice versa) to be negligible. Moreover, we assume that the wrapper adversary responds to the wrapped adversary with some fixed value immediately. That is, we consider the only overhead associated with additional computations performed by the wrapper adversary.

In case of $p = 0$, security of the CS_{SAN} scheme is fully described by security of corresponding AE_{SAN} scheme. More

precisely, the following Theorem holds.

Theorem 1 (CS_{SAN} security, case $p = 0$). Let MODEL-ATK be a relevant provable security model. Given the $\text{CS}_{\text{SAN}} = \text{CS}_{\text{SAN}}[\tilde{\mathcal{E}}, \text{MAC}, \text{KDF}, 0]$ scheme and corresponding $\text{AE}_{\text{SAN}} = \text{AE}_{\text{SAN}}[\tilde{\mathcal{E}}, \text{MAC}]$ scheme, the following statement holds:

$$\text{Adv}_{\text{CS}_{\text{SAN}}}^{\text{MODEL-ATK}}(t, a_1, \dots, a_u) \leq \text{Adv}_{\text{AE}_{\text{SAN}}}^{\text{MODEL-ATK}}(t, a_1, \dots, a_u). \quad (2)$$

where $\{t, a_1, \dots, a_u\}$ is some set of adversary resources bounds.

Informally, Theorem 1 states, that if the AE_{SAN} scheme is secure in MODEL-ATK setting (in particular, in the *RND-adeCCA-sector model*), then corresponding CS_{SAN} scheme (in case of p being equal to 0) is equally secure in the same setting. Now we proceed to the proof.

Proof of Theorem 1. The proof idea is as follows: given an arbitrary adversary $\mathcal{A} \in \mathcal{A}(t, a_1, \dots, a_u)$, we build on top of it an adversary $\mathcal{B}$ and prove that its resources are bounded by the same set $\{t, a_1, \dots, a_u\}$. We would use such proof technique many times throughout this paper.

In fact, the adversary $\mathcal{B}$ simply acts as an oracle for the adversary $\mathcal{A}$ forwarding $\mathcal{A}$ ’s queries q_i to its own oracle and oracle’s responses r_i back to $\mathcal{A}$, $i = \overline{1, q}$, where q is a total number of $\mathcal{A}$ ’s queries. After processing all queries, the adversary $\mathcal{B}$ returns the result of the adversary $\mathcal{A}$ as its own. Such adversary $\mathcal{B}$ is shown in Fig. 10.

By definition of the adversary $\mathcal{B}$, we have:

$$\text{Adv}_{\text{CS}_{\text{SAN}}}^{\text{MODEL-ATK}}(\mathcal{A}) \leq \text{Adv}_{\text{AE}_{\text{SAN}}}^{\text{MODEL-ATK}}(\mathcal{B}). \quad (3)$$

Clearly, the adversary $\mathcal{B}$ does not perform any additional computations. Hence, $\mathcal{B}$ performs in the same time as $\mathcal{A}$ does. Furthermore, the $\mathcal{B}$ adversary does not impose any other restrictions. Therefore, the set of resources bounds $\{t, a_1, \dots, a_u\}$ is valid for $\mathcal{B}$. By definition, we have:

$$\text{Adv}_{\text{AE}_{\text{SAN}}}^{\text{MODEL-ATK}}(\mathcal{B}) \leq \text{Adv}_{\text{AE}_{\text{SAN}}}^{\text{MODEL-ATK}}(t, a_1, \dots, a_u). \quad (4)$$

Finally, since the adversary $\mathcal{A}$ is arbitrary one from the set $\mathcal{A}(t, a_1, \dots, a_u)$, we obtain the Theorem statement from (3) and (4). **Q.E.D.**

Case $p > 0$. To analyze security of the $\text{CS}_{\text{SAN}}[\tilde{\mathcal{E}}, \text{MAC}, \text{KDF}, p]$ scheme with p being greater than 0, we use hybrid argument. In this case, security of the whole scheme depends on security of each component and could not be directly reduced to security of corresponding AE_{SAN} scheme only. For simplicity, we consider writing into a single storage object, that is, ON is fixed.

Theorem 2 (CS_{SAN} security, case $p > 0$). Let $\text{CS}_{\text{SAN}} =$

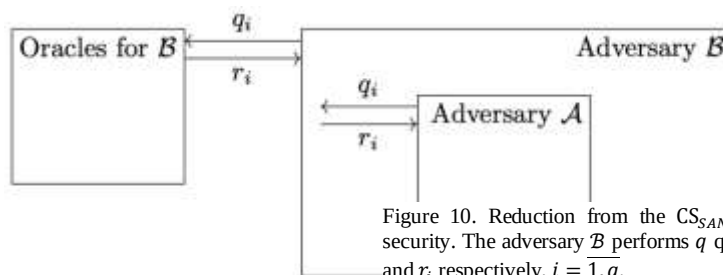


Figure 10. Reduction from the CS_{SAN} scheme (with $p = 0$) security to the AE_{SAN} scheme security. The adversary $\mathcal{B}$ performs q queries in total, i -th query and response are denoted by q_i and r_i respectively, $i = \overline{1, q}$.

$\text{CS}_{\text{SAN}}[\mathcal{E}, \text{MAC}, \text{KDF}, p]$ for $p > 0$ and $\text{AE}_{\text{SAN}} = \text{AE}_{\text{SAN}}[\mathcal{E}, \text{MAC}]$. Let n be a fixed positive integer. The following inequality holds:

$$\begin{aligned} & \text{Adv}_{\text{CS}_{\text{SAN}}}^{\text{RND-adeCCA-sector}}(t, q_e, q_m) \\ & \leq \text{Adv}_{\text{KDF}}^{\text{VO-PRF}}(t_k, q_k) \\ & + \sum_{j=0}^{2^p-1} \text{Adv}_{\text{AE}_{\text{SAN}}^{(j)}}^{\text{RND-adeCCA-sector}}(t_j, q_{e,j}, q_{m,j}). \end{aligned} \quad (5)$$

where $\text{AE}_{\text{SAN}}^{(j)}$ is an authenticated encryption scheme AE_{SAN} instantiated using j -th pair of encryption and authentication keys, and:

- $q_k = 2^p$;
- $t_k \leq t + \alpha_{k,1} \cdot 2^p + \alpha_{k,2} \cdot (q_e + q_m)n$, $\alpha_{k,1}, \alpha_{k,2} \in \mathbb{R}^+$;
- $\sum_{j=0}^{2^p-1} q_{e,j} = q_e$;
- $\sum_{j=0}^{2^p-1} q_{m,j} = q_m$;
- $t_j \leq t + \alpha_{j,1} \cdot 2^p + \alpha_{j,2} \cdot (q_e + q_m)n$, $\alpha_{j,1}, \alpha_{j,2} \in \mathbb{R}^+$, $j = \overline{0, 2^p-1}$.

Note that in case of $\mathcal{K}_e = \mathcal{K}_m$, the VO-PRF model could be replaced by the PRF model, because all KDF outputs would have the same length.

Proof of Theorem 2. The idea of the proof is as follows. First, we replace KDF with a random function. By doing this, we get 2^p independent subsystems $\text{AE}_{\text{SAN}}^{(j)}$, $j = \overline{0, 2^p-1}$. Finally, we apply hybrid argument by replacing each of them by ideal one.

We consider a sequence of security games CS , $\text{CS\$}$, $\text{CS\$}_0$, ..., $\text{CS\$}_{2^p-1}$. In each game we slightly modify the CS_{SAN} scheme.

In the CS game, the adversary interacts with the CS_{SAN} scheme in its original form as defined in the Section 3.

The $\text{CS\$}$ game differs from the CS game in generation of encryption and authentication keys. Namely, in the $\text{CS\$}$ game, key derivation function of the CS_{SAN} scheme is replaced by a random function. We denote such modified scheme by $\text{CS\$}_{\text{SAN}}$. To make further statements mathematically correct, we also use notation $\text{CS\$}_{-1}$ and $\text{CS\$}_{\text{SAN}}^{-1}$ to denote the CS game and the $\text{CS\$}_{\text{SAN}}$ scheme respectively.

The $\text{CS\$}_j$ game is similar to the $\text{CS\$}_{j-1}$ game except of using ideal deterministic authenticated encryption scheme instead of $\text{AE}_{\text{SAN}}^{(j)}$, $j = \overline{0, 2^p-1}$. That is, in the $\text{CS\$}_j$ game all $\text{AE}_{\text{SAN}}^{(v)}$ schemes for $v = \overline{0, j}$ are replaced by ideal ones. We denote modified CS_{SAN} scheme in the $\text{CS\$}_j$ game by $\text{CS}_{\text{SAN}}^{\$,j}$.

Without loss of generality, we focus on adversaries such that their advantages are non-negative. Consider an arbitrary adversary $\mathcal{A} \in \mathcal{A}(t, q_e, q_m)$. Our goal is to upper bound its advantage. The following set of equalities holds:

$$\begin{aligned} & \text{Adv}_{\text{CS}_{\text{SAN}}}^{\text{RND-adeCCA-sector}}(\mathcal{A}) \\ & = \Pr[\mathcal{A}^{\text{CS}_{\text{SAN}}} \Rightarrow 1] - \Pr[\mathcal{A}^{(\$,\perp)} \Rightarrow 1] \\ & = \Pr[\mathcal{A}^{\text{CS}_{\text{SAN}}} \Rightarrow 1] - \sum_{j=0}^{2^p-1} \Pr[\mathcal{A}^{\text{CS}_{\text{SAN}}^{\$,j}} \Rightarrow 1] \end{aligned}$$

$$\begin{aligned} & + \sum_{j=0}^{2^p-1} \Pr[\mathcal{A}^{\text{CS}_{\text{SAN}}^{\$,j}} \Rightarrow 1] - \Pr[\mathcal{A}^{(\$,\perp)} \Rightarrow 1] \\ & = \text{Adv}_{\text{CS}_{\text{SAN}}, \text{CS}_{\text{SAN}}^{\$,j}}^{\text{IND}}(\mathcal{A}) \\ & + \sum_{j=0}^{2^p-1} \text{Adv}_{\text{CS}_{\text{SAN}}^{\$,j-1}, \text{CS}_{\text{SAN}}^{\$,j}}^{\text{IND}}(\mathcal{A}). \end{aligned} \quad (6)$$

where the last equality holds due to $\Pr[\mathcal{A}^{(\$,\perp)} \Rightarrow 1] = \Pr[\mathcal{A}^{\text{CS}_{\text{SAN}}^{\$,2^p-1}} \Rightarrow 1]$, which is true by definition of the $\text{CS}_{\text{SAN}}^{\$,2^p-1}$ game. It is necessary to emphasize, that we could use hybrid argument, because keys are independent in the $\text{CS\$}$ game, that is, separate encryption schemes $\text{AE}_{\text{SAN}}^{(j)}$ are independent, $j = \overline{0, 2^p-1}$.

Now our goal is to upper bound distinguishing advantages between games CS , $\text{CS\$}$, $\text{CS\$}_0$, ..., $\text{CS\$}_{2^p-1}$.

First, we build an adversary $\mathcal{B}$, such that the following inequality holds and then estimate its resources (time and number of queries):

$$\text{Adv}_{\text{CS}_{\text{SAN}}, \text{CS}_{\text{SAN}}^{\$,j}}^{\text{IND}}(\mathcal{A}) \leq \text{Adv}_{\text{KDF}}^{\text{VO-PRF}}(\mathcal{B}). \quad (7)$$

Indeed, the CS_{SAN} and $\text{CS}_{\text{SAN}}^{\$,j}$ schemes differ only in key derivation function. If these schemes could be distinguished, then we could use this fact to distinguish between KDF and a random function. The adversary $\mathcal{B}$ wraps $\mathcal{A}$ as follows. Before handling queries from $\mathcal{A}$, it uniformly and independently chooses master keys K_e^{ON} and K_m^{ON} , and invokes the CS_{SAN} -KeyGen algorithm, forwarding each KDF invocation to its oracle. Then, the adversary $\mathcal{B}$ simulates CS_{SAN} scheme while handling queries from $\mathcal{A}$. After $\mathcal{A}$ returns a value b' , $\mathcal{B}$ returns the same value.

Next, we estimate upper bounds on $\mathcal{B}$ resources. Later in the article we only specify a final estimation omitting intermediate steps. Clearly, $\mathcal{B}$ performs exactly $q_k = 2^p$ queries to its oracle. Each query takes at most $\alpha_{k,1} \in \mathbb{R}^+$ time units. On each of $q_e + q_m$ queries from $\mathcal{A}$, $\mathcal{B}$ processes exactly n blocks of data. Processing of each block takes at most $\alpha_{k,2} \in \mathbb{R}^+$ time units. In total, $\mathcal{B}$ performs in time $t_k \leq t + \alpha_{k,1} \cdot 2^p + \alpha_{k,2} \cdot (q_e + q_m)n$, where term t appears, because $\mathcal{B}$ does not return until $\mathcal{A}$ returns.

If $\mathcal{B}$ interacts with KDF (random function), then $\mathcal{A}$ interacts with CS_{SAN} ($\text{CS}_{\text{SAN}}^{\$,j}$). Therefore, the inequality (7) holds. Now, by definition we have:

$$\text{Adv}_{\text{KDF}}^{\text{VO-PRF}}(\mathcal{B}) \leq \text{Adv}_{\text{KDF}}^{\text{VO-PRF}}(t_k, q_k). \quad (8)$$

The $\text{CS}_{\text{SAN}}^{\$,j}$ scheme contains 2^p independent subsystems, each with its own authenticated encryption scheme $\text{AE}_{\text{SAN}}^{(j)}$, $j = \overline{0, 2^p-1}$. Consider a pair of games $\text{CS\$}_{j-1}$ and $\text{CS\$}_j$. These games differ only in j -th subsystem: oracles in the former game use “real” subsystem AE_{SAN} , oracles in the latter game use “ideal” subsystem $(\$,\perp)$. Now we build an adversary $\mathcal{B}_j$, such that the following inequality holds:

$$\text{Adv}_{\text{CS}_{\text{SAN}}^{\$,j-1}, \text{CS}_{\text{SAN}}^{\$,j}}^{\text{IND}}(\mathcal{A}) \leq \text{Adv}_{\text{AE}_{\text{SAN}}^{(j)}}^{\text{RND-adeCCA-sector}}(\mathcal{B}_j). \quad (9)$$

We use similar idea as before. The adversary $\mathcal{B}_j$ before handling queries from $\mathcal{A}$ uniformly and independently generates keys for subsystems j' , where $j < j' \leq 2^p-1$. Upon receiving a query from $\mathcal{A}$, the adversary $\mathcal{B}_j$ decides

which subsystem to interact with based on a unit number UN using p most significant bits of it (exact decision algorithm could be easily figured out based on the algorithm in Fig. 6). Let this subsystem has number j' . The following cases could occur:

- $j' < j$: $\mathcal{B}_j$ invokes ideal oracles $\$$ or $\perp$ to produce responses (due to definition of the RND-adeCCA-sector model, the adversary $\mathcal{B}_j$ does not need to maintain lists of responses);
- $j' > j$: $\mathcal{B}_j$ invokes encryption or decryption function of the $\text{AE}_{\text{SAN}}^{(j')}$ scheme to produce responses;
- $j' = j$: $\mathcal{B}_j$ forwards queries to its own oracles that implement either $\text{AE}_{\text{SAN}}^{(j)}$ or $(\$, \perp)$, and sends their responses back to $\mathcal{A}$.

After handling all queries from $\mathcal{A}$, the adversary $\mathcal{B}_j$ returns a value returned by $\mathcal{A}$.

The adversary $\mathcal{B}_j$ makes $q_{e,j}$ ($q_{m,j}$) queries to its own “left” (“right”) oracle, where $\sum_{j=0}^{2^p-1} q_{e,j} = q_e$ ($\sum_{j=0}^{2^p-1} q_{m,j} = q_m$), and performs in time $t_j \leq t + \alpha_{j,1} \cdot 2^p + \alpha_{j,2} \cdot (q_{e,j} + q_{m,j})n$.

If $\mathcal{B}_j$ interacts with “real” (“ideal”) oracles, then it simulates $\text{CS}_{\text{SAN}}^{\$,j-1}$ ($\text{CS}_{\text{SAN}}^{\$,j}$) for $\mathcal{A}$. Therefore, inequality (9) holds. By definition, we have:

$$\begin{aligned} \text{Adv}_{\text{AE}_{\text{SAN}}^{(j)}}^{\text{RND-adeCCA-sector}}(\mathcal{B}_j) \\ \leq \text{Adv}_{\text{AE}_{\text{SAN}}^{(j)}}^{\text{RND-adeCCA-sector}}(t_j, q_{e,j}, q_{m,j}). \end{aligned} \quad (10)$$

The Theorem statement follows from (6) – (10) given $\mathcal{A}$ being an arbitrary adversary from $\mathcal{A}(t, q_e, q_m)$. **Q.E.D.**

V. UNDERLYING ALGORITHMS REQUIREMENTS

In this section, we provide security requirements to underlying algorithms of the CS_{SAN} scheme. These requirements are provided by reductions of the RND-adeCCA-sector model to other models.

There are three methods of combining confidentiality and integrity block cipher modes of operation into a single symmetric cryptosystem (generic compositions). These methods are called Encrypt-then-MAC (EtM), MAC-then-Encrypt (MtE), and Encrypt-and-MAC (E&M) constructions [26]. In [26], Bellare and Namprempre provide security analysis of these constructions for “conventional” confidentiality modes. In [32], Namprempre et al. consider more variations of generic compositions for nonce-based and IV-based encryption schemes. In this section, we show security of the modified EtM construction (because this construction is used in the CS_{SAN} scheme) with a tweakable encryption scheme (e.g. the XEH mode of operation [11]).

Let $\tilde{\mathcal{E}}$ and MAC be a tweakable encryption scheme and a message authenticated code scheme respectively. We denote the modified EtM construction built on top of these schemes by $\text{EtM}^*[\tilde{\mathcal{E}}, \text{MAC}]$. Encryption function of this construction produces ciphertext c by invocation of the $\tilde{\mathcal{E}}$ scheme encryption function and then computes a tag tag of ciphertext c concatenated with a tweak. Decryption function verifies a tag and then in case of tag correctness produces a plaintext by decrypting it using decryption function of the $\tilde{\mathcal{E}}$

scheme. If the tag is incorrect, then the decryption function returns $\perp$. Clearly, $\text{AE}_{\text{SAN}}[\tilde{\mathcal{E}}, \text{MAC}] = \text{EtM}^*[\tilde{\mathcal{E}}, \text{MAC}]$ (one could verify this fact by referring to Fig. 8 and Fig. 9).

The crucial part of the modified EtM construction is to produce the tag not from the ciphertext only, but also from the tweak. Otherwise, a trivial forgery attack is possible. Indeed, the adversary copies ciphertext from one sector to another and replaces corresponding tag. In this case tag would be verified successfully in another sector. Namely, the following statement holds.

Statement 1 (forgery attack on “conventional” EtM construction). Let $\tilde{\mathcal{E}}$ and MAC be a tweakable encryption scheme and a message authenticated code scheme respectively. Let $\text{EtM}[\tilde{\mathcal{E}}, \text{MAC}]$ be a (conventional) Encrypt-then-MAC construction built on top of $\tilde{\mathcal{E}}$ and MAC. If ciphertext c and tag tag are produced using tweak $T_1 \in \mathcal{T}_{\text{tw}}$, then c could be decrypted (in another plaintext) using tweak $T_2 \neq T_1$.

Proof. The statement directly follows from the following fact: $\text{tag} = S(K, c)$, where K is a MAC key, S is a signing function. That is, tweak is not used by signing function. Hence, the tweak is not used by verification function either. Therefore, equality $V(K, c, \text{tag}) = 1$ holds for every tweak. **Q.E.D.**

The modified EtM construction is similar to the N2 construction in [32], but does not require nonce.

The intuition behind the modified EtM construction security reduction is as follows. Queries to the “right” oracle are “covered” by secure MAC scheme, and queries to the “left” oracle of the RND-adeCCA-sector model are “covered” by secure tweakable encryption scheme. Security properties of the EtM construction are summarized in the following Theorem.

Theorem 3 (Modified EtM construction security). Let $\tilde{\mathcal{E}}$ and MAC be a tweakable encryption scheme and a message authenticated code scheme respectively. The following inequality holds:

$$\begin{aligned} \text{Adv}_{\text{EtM}^*[\tilde{\mathcal{E}}, \text{MAC}]}^{\text{RND-adeCCA-sector}}(t, q_e, q_m) \\ \leq \text{Adv}_{\tilde{\mathcal{E}}}^{\text{RND-fdeCPA-sector}}(t_e, q_e) \\ + \text{Adv}_{\text{MAC}}^{\text{PRF}}(t_m, q_e) \\ + q_m \cdot \text{Adv}_{\text{MAC}}^{\text{SUF-CMA}}(t, q_e), \end{aligned}$$

where:

- $t_e = t + \alpha_e \cdot q_e$, $\alpha_e \in \mathbb{R}^+$;
- $t_m = t + \alpha_m \cdot q_m$, $\alpha_m \in \mathbb{R}^+$.

Before proving the Theorem 3, we provide useful Lemmas. The first one binds INT-adeCTXT-sector and INT-adeCTXT1-sector security notions. The second one reduces RND-adeCCA-sector security of a DAE scheme to its INT-adeCTXT-sector and RND-fdeCPA-sector security. We use these Lemmas as intermediate steps in the Theorem 3 proof.

Lemma 1. Let $\tilde{\mathcal{E}}$ be a deterministic authenticated encryption scheme. Then the following inequality holds:

$$\text{Adv}_{\mathcal{E}}^{\text{INT-adeCTXT-sector}}(t, q_e, q_m) \leq q_m \cdot \text{Adv}_{\mathcal{E}}^{\text{INT-adeCTXT1-sector}}(t, q_e). \quad (11)$$

Proof sketch. The lemma is proven merely by straightforward applying of hybrid argument over decryption queries. We provide the proof idea.

Let $\mathcal{A}$ be an adversary against $\mathcal{E}$ in the INTadeCTXT-sector game. The adversary $\mathcal{B}$ is built on top of $\mathcal{A}$ as follows. Before handling queries, $\mathcal{B}$ uniformly chooses $j' \in \mathbb{N}^{\leq q_m}$, i.e. makes a guess which query is a forgery. Then it forwards encryption queries to own encryption oracle and rejects all decryption forgery except for the query with number j' . Clearly, $\mathcal{B}$ correctly guesses forgery attempt with probability at least q_m^{-1} . That is why q_m factor appears in (11).

Lemma 2. Let $\mathcal{E}$ be a deterministic authenticated encryption scheme. Then the following inequality holds:

$$\begin{aligned} \text{Adv}_{\mathcal{E}}^{\text{RND-adeCCA-sector}}(t, q_e, q_m) &\leq \text{Adv}_{\mathcal{E}}^{\text{RND-fdeCPA-sector}}(t, q_e) \\ &\quad + q_m \cdot \text{Adv}_{\mathcal{E}}^{\text{INT-CTXT1-sector}}(t, q_e). \end{aligned}$$

Proof. Let $\mathcal{A} \in \mathcal{A}(t, q_e, q_m)$ be an adversary in the RND-adeCCA-sector game against $\mathcal{E}$ scheme. We have:

$$\begin{aligned} \text{Adv}_{\mathcal{E}}^{\text{RND-adeCCA-sector}}(\mathcal{A}) &= \Pr[\mathcal{A}^{(\tilde{E}, \tilde{E}^{-1})} \Rightarrow 1] - \Pr[\mathcal{A}^{(\$, \perp)} \Rightarrow 1] \\ &= (\Pr[\mathcal{A}^{(\tilde{E}, \tilde{E}^{-1})} \Rightarrow 1] - \Pr[\mathcal{A}^{(\tilde{E}, \perp)} \Rightarrow 1]) \\ &\quad + (\Pr[\mathcal{A}^{(\tilde{E}, \perp)} \Rightarrow 1] - \Pr[\mathcal{A}^{(\$, \perp)} \Rightarrow 1]) \\ &= \text{Adv}_{(\tilde{E}, \tilde{E}^{-1}), (\tilde{E}, \perp)}^{\text{IND}}(\mathcal{A}) + \text{Adv}_{(\tilde{E}, \perp), (\$, \perp)}^{\text{IND}}(\mathcal{A}). \end{aligned} \quad (12)$$

The first summand in (12) could be upper bounded as follows. The adversary has to distinguish between two experiment forms that differ only in a way how decryption oracle responds to the adversary. That is, in the “real” world decryption oracle responds with decryption result, in the “random” world always rejects ciphertexts. These experiment forms are identical, until the adversary submits a ciphertext such that the decryption oracle in the “real” world accepts it and returns a valid plaintext (i.e. not $\perp$). That is, given an event of ciphertext forgery denoted by CF , we have:

$$\text{Adv}_{(\tilde{E}, \tilde{E}^{-1}), (\tilde{E}, \perp)}^{\text{IND}}(\mathcal{A}) \leq \Pr[CF]. \quad (13)$$

To upper bound the value of $\Pr[CF]$ we build an adversary $\mathcal{C}$ aiming to win INT-adeCTXT-sector game. The adversary $\mathcal{C}$ is built on top of $\mathcal{A}$ and forwards queries from $\mathcal{A}$ to its own oracles without changes. Responses of the oracles are forwarded back to $\mathcal{A}$ without changes too. The adversary $\mathcal{C}$ wins the INT-adeCTXT-sector game if and only if $\mathcal{A}$ forges a ciphertext. That is, by definition we have:

$$\begin{aligned} \Pr[CF] &\leq \text{Adv}_{\mathcal{E}}^{\text{INT-adeCTXT-sector}}(\mathcal{C}) \\ &\leq \text{Adv}_{\mathcal{E}}^{\text{INT-adeCTXT-sector}}(t, q_e, q_m). \end{aligned} \quad (14)$$

To upper bound the second summand in (12), we build an adversary $\mathcal{B}$ such that:

$$\text{Adv}_{(\tilde{E}, \perp), (\$, \perp)}^{\text{IND}}(\mathcal{A}) \leq \text{Adv}_{\mathcal{E}}^{\text{RND-fdeCPA-sector}}(\mathcal{B}). \quad (15)$$

The adversary $\mathcal{B}$ wraps the adversary $\mathcal{A}$. Encryption queries are forwarded to $\mathcal{B}$ oracle, responses are sent back to $\mathcal{A}$. On decryption queries $\mathcal{B}$ responds with $\perp$. Note, that

decryption queries do not provide any information to the adversary and could be eliminated without loss of generality. The result returned by $\mathcal{A}$ is returned by $\mathcal{B}$ without any changes.

Clearly, $\mathcal{B}$ performs in the same time as $\mathcal{A}$ does. Further, $\mathcal{B}$ makes exactly q_e queries to its own oracle. Hence, the following holds:

$$\text{Adv}_{\mathcal{E}}^{\text{RND-fdeCPA-sector}}(\mathcal{B}) \leq \text{Adv}_{\mathcal{E}}^{\text{RND-fdeCPA-sector}}(t, q_e). \quad (16)$$

We obtain the Lemma statement from (12) – (16) given Lemma 1. **Q.E.D.**

Lemma 3. Let $F: X \rightarrow Y$ be a function uniformly chosen from $F(X, Y)$ and let $\$$ be a random oracle. Let $\{x_1, \dots, x_n\}$ be a sequence of distinct elements of X . Consider two sequences: $\mathfrak{T}_1 = \{F(x_1), \dots, F(x_n)\}$ and $\mathfrak{T}_2 = \{\$(x_1), \dots, \$(x_n)\}$. Corresponding values in these sequences are equally distributed.

Informally, Lemma 3 states that random function and random oracle “look” the same, until there is no repetition in their inputs.

Proof of Lemma 3. Let’s consider $\mathfrak{T}_2$ first. Since $\$$ is a random oracle, all values in $\mathfrak{T}_2$ are uniformly and independently distributed on V_l .

Random function F could be simulated using lazy sampling procedure. That is, on each new input a new output value is uniformly sampled from Y and stored in a table. On repeated input, corresponding stored value is returned. Since all inputs are distinct, all values in $\mathfrak{T}_1$ are uniformly and independently distributed. This proves the Lemma. **Q.E.D.**

Now we proceed to the Theorem 3 proof.

Proof of the Theorem 3. This proof consists of two reductions. The $\text{EtM}^*[\mathcal{E}, \text{MAC}]$ scheme INT-adeCTXT1-sector security is reduced to the MAC scheme SUF-CMA security. The $\text{EtM}^*[\mathcal{E}, \text{MAC}]$ scheme RND-fdeCPA-sector security is reduced to the $\mathcal{E}$ scheme RND-fdeCPA-sector security. The Theorem statement follows directly from these reductions and the Lemma 2.

Reduction 1. The first reduction could be easily shown. Let $\mathcal{A}_{\text{int}} \in \mathcal{A}(t, q_e)$ be an adversary against $\text{EtM}^*[\mathcal{E}, \text{MAC}]$ scheme in the INT-adeCTXT1-sector game. We build adversary $\mathcal{B}$ such that the inequality holds:

$$\text{Adv}_{\text{EtM}^*[\mathcal{E}, \text{MAC}]}^{\text{INT-adeCTXT1-sector}}(\mathcal{A}_{\text{int}}) \leq \text{Adv}_{\text{MAC}}^{\text{SUF-CMA}}(\mathcal{B}). \quad (17)$$

The adversary $\mathcal{B}$ plays role of oracles for the adversary $\mathcal{A}_{\text{int}}$. In the beginning, $\mathcal{B}$ uniformly samples key for the $\mathcal{E}$ scheme. Whenever $\mathcal{A}_{\text{int}}$ performs an encryption query (T_{tw}, m) , $\mathcal{B}$ encrypts message m obtaining ciphertext c and makes query $(c || T_{\text{tw}})$ to its first oracle and returns the response back to $\mathcal{A}_{\text{int}}$. If $\mathcal{A}_{\text{int}}$ performs decryption query $(T_{\text{tw}}, c, \text{tag})$ (forgery), then $\mathcal{B}$ performs forgery attempt $(c || T_{\text{tw}}, \text{tag})$. Since $\mathcal{A}_{\text{int}}$ performs valid forgery attempts, forgery attempts of $\mathcal{B}$ are also valid due to injectivity of $(T_{\text{tw}}, c) \rightarrow c || T_{\text{tw}}$ mapping.

The adversary $\mathcal{B}$ makes exactly q_e queries to the oracle and performs in the same time as $\mathcal{A}_{\text{int}}$ does. We consider a single transformation $(T_{\text{tw}}, c, \text{tag}) \rightarrow (c || T_{\text{tw}}, \text{tag})$ to be

performed in negligible time. By definition, the following inequality holds:

$$\text{Adv}_{\text{MAC}}^{\text{SUF-CMA}}(\mathcal{B}) \leq \text{Adv}_{\text{MAC}}^{\text{SUF-CMA}}(t, q_e). \quad (18)$$

Reduction 2. The second reduction consists of two parts on its own. For an adversary $\mathcal{A}_{cpa} \in \mathcal{A}(t, q_e)$ against the $\text{EtM}^*[\tilde{\mathcal{E}}, \text{MAC}]$ scheme in the RND-fdeCPA-sector game we have:

$$\begin{aligned} & \text{Adv}_{\text{EtM}^*[\tilde{\mathcal{E}}, \text{MAC}]}^{\text{RND-fdeCPA-sector}}(\mathcal{A}_{cpa}) \\ &= \Pr[\mathcal{A}_{cpa}^{\text{EtM}^*[\tilde{\mathcal{E}}, \text{MAC}]} \Rightarrow 1] - \Pr[\mathcal{A}_{cpa}^{\$} \Rightarrow 1] \\ &= \left(\Pr[\mathcal{A}_{cpa}^{\text{EtM}^*[\tilde{\mathcal{E}}, \text{MAC}]} \Rightarrow 1] - \Pr[\mathcal{A}_{cpa}^{\text{EtM}^*[\tilde{\mathcal{E}}]} \Rightarrow 1] \right) \\ &\quad - \left(\Pr[\mathcal{A}_{cpa}^{\text{EtM}^*[\tilde{\mathcal{E}}]} \Rightarrow 1] - \Pr[\mathcal{A}_{cpa}^{\$} \Rightarrow 1] \right) \\ &= \text{Adv}_{\text{EtM}^*[\tilde{\mathcal{E}}, \text{MAC}], \text{EtM}^*[\tilde{\mathcal{E}}]}^{\text{IND}}(\mathcal{A}_{cpa}) + \text{Adv}_{\text{EtM}^*[\tilde{\mathcal{E}}], \$}^{\text{IND}}(\mathcal{A}_{cpa}), \end{aligned} \quad (19)$$

where $\text{EtM}^*[\tilde{\mathcal{E}}]$ is a DAE scheme built from the $\text{EtM}^*[\tilde{\mathcal{E}}, \text{MAC}]$ scheme by replacing the MAC scheme with a truly random function.

First, we build an adversary $\mathcal{C}$ such that its advantage against the $\tilde{\mathcal{E}}$ scheme in the RND-fdeCPA-sector game upper bounds the advantage of the $\mathcal{A}_{cpa}$ adversary that distinguishes between the $\text{EtM}^*[\tilde{\mathcal{E}}]$ scheme and ideal oracle $\$,$ i.e. the second summand in (19):

$$\text{Adv}_{\text{EtM}^*[\tilde{\mathcal{E}}], \$}^{\text{IND}}(\mathcal{A}_{cpa}) \leq \text{Adv}_{\tilde{\mathcal{E}}}^{\text{RND-fdeCPA-sector}}(\mathcal{C}). \quad (20)$$

The adversary $\mathcal{C}$ forwards encryption queries of the adversary $\mathcal{A}_{cpa}$ to its own encryption oracle, receives ciphertext c , uniformly samples tag tag from the tag space, and sends a pair (c, tag) back to $\mathcal{A}_{cpa}$. After handling all queries, $\mathcal{C}$ returns whatever the adversary $\mathcal{A}_{cpa}$ returns. Hence, the inequality (20) holds.

Here is a tricky point: the $\text{EtM}^*[\tilde{\mathcal{E}}]$ scheme uses random function to produce tags, but the adversary $\mathcal{C}$ samples tags uniformly and independently on each query, simulating a random oracle. These are different concepts. Let's handle this explicitly. By definition of the RND-fdeCPA-sector model, the adversary $\mathcal{A}_{cpa}$ never repeats its queries, i.e. all pairs (m, T) are distinct. Hence, all $c||T$ values are distinct. By Lemma 3, we could use random oracle and random function interchangeably in this context. This relaxes requirements for the MAC scheme, because otherwise it would be necessary for MAC to be secure in random oracle model.

The adversary $\mathcal{C}$ makes exactly q_e queries to its oracle and performs in time $t_e = t + \alpha_e \cdot q_e$, where $\alpha_e \in \mathbb{R}^+$. Therefore, the inequality holds:

$$\text{Adv}_{\tilde{\mathcal{E}}}^{\text{RND-fdeCPA-sector}}(\mathcal{C}) \leq \text{Adv}_{\tilde{\mathcal{E}}}^{\text{RND-fdeCPA-sector}}(t_e, q_e). \quad (21)$$

To upper bound the first summand in (19), we build an adversary $\mathcal{D}$ on top of $\mathcal{A}_{cpa}$ such that the following inequality holds:

$$\text{Adv}_{\text{EtM}^*[\tilde{\mathcal{E}}, \text{MAC}], \text{EtM}^*[\tilde{\mathcal{E}}]}^{\text{IND}}(\mathcal{A}_{cpa}) \leq \text{Adv}_{\text{MAC}}^{\text{PRF}}(\mathcal{D}). \quad (22)$$

The adversary $\mathcal{D}$ plays role of oracle for $\mathcal{A}_{cpa}$. It uniformly samples an encryption key. On encryption query from $\mathcal{A}_{cpa}$, the adversary $\mathcal{D}$ encrypts the received message using received tweak, performs query to its own oracle to compute tag, and returns ciphertext-tag pair to $\mathcal{A}_{cpa}$. After handling all queries, it returns whatever the adversary $\mathcal{A}_{cpa}$ returns. Therefore, inequality (22) holds. The adversary $\mathcal{D}$ makes exactly q_e queries to the oracle and performs in time

$t_m = t + \alpha_m \cdot q_e n$, where $\alpha_m \in \mathbb{R}^+$. Hence, the adversary $\mathcal{D}$ advantage is upper bounded:

$$\text{Adv}_{\text{MAC}}^{\text{PRF}}(\mathcal{D}) \leq \text{Adv}_{\text{MAC}}^{\text{PRF}}(t_m, q_e). \quad (23)$$

By Lemma 2, given (17) – (23) we obtain the Theorem statement. **Q.E.D.**

Corollary 1. Let $\tilde{\mathcal{E}}$ and MAC be a tweakable encryption scheme and a message authenticated code scheme respectively. The following inequality holds:

$$\begin{aligned} & \text{Adv}_{\text{EtM}^*[\tilde{\mathcal{E}}, \text{MAC}]}^{\text{RND-adeCCA-sector}}(t, q_e, q_m) \\ & \leq \text{Adv}_{\tilde{\mathcal{E}}}^{\text{RND-fdeCCA-sector}}(t_e, q_e, 0) \\ & \quad + \text{Adv}_{\text{MAC}}^{\text{PRF}}(t_m, q_e) \\ & \quad + q_m \cdot \text{Adv}_{\text{MAC}}^{\text{SUF-CMA}}(t, q_e), \end{aligned}$$

where:

- $t_e = t + \alpha_e \cdot q_e, \alpha_e \in \mathbb{R}^+;$
- $t_m = t + \alpha_m \cdot q_m, \alpha_m \in \mathbb{R}^+.$

Proof sketch. To prove this corollary, it is enough to note that any adversary in the RND-fdeCPA-sector game could be used in RND-fdeCCA-sector game performing zero decryption queries.

The following Lemma binds a block cipher-based tweakable encryption scheme security and a block cipher security. This Lemma is a crucial part in reduction of a composite scheme security to a basic problem complexity. In our case, we use indistinguishability of the block cipher (and its inverse) from a random permutation (and its inverse) as the basic problem.

Lemma 4. Let $\mathcal{E}$ be a block cipher, $\tilde{\mathcal{E}} = \text{WBE}[\mathcal{E}]$ be a tweakable encryption scheme built on top of the block cipher $\mathcal{E}$ using wide block encryption mode of operation WBE. Suppose the $\tilde{\mathcal{E}}$ scheme requires k independent block cipher keys $K_0, \dots, K_{k-1}$. Let the K_j key is used $v_{ee,j}$ times for encryption, $v_{de,j}$ times for decryption in encryption function of the $\tilde{\mathcal{E}}$ scheme, and $v_{ed,j}$ times for encryption, $v_{dd,j}$ times for decryption in decryption function, $j = \overline{0, k-1}$. Let $\text{WBE}[\pi_0, \dots, \pi_{k-1}]$ be a tweakable encryption scheme built from $\tilde{\mathcal{E}}$ by replacing each invocation of E_{K_j} ($E_{K_j}^{-1}$) by π_j (π_j^{-1}), where π_j is a random permutation chosen uniformly and independently, $j = \overline{0, k-1}$. Then the following relation holds:

$$\begin{aligned} & \text{Adv}_{\text{WBE}[\tilde{\mathcal{E}}]}^{\text{RND-fdeCCA-sector}}(t, q_e, q_d) \\ & \leq \text{Adv}_{\text{WBE}[\pi_0, \dots, \pi_{k-1}]}^{\text{RND-fdeCCA-sector}}(\infty, q_e, q_d) \\ & \quad + \sum_{j=0}^{k-1} \text{Adv}_{\tilde{\mathcal{E}}}^{\pm \text{PRP}}(t_j, q_{e,j}, q_{d,j}), \end{aligned} \quad (24)$$

where:

- $q_{e,j} = q_e \cdot v_{ee,j} + q_d \cdot v_{ed,j}, j = \overline{0, k-1};$
- $q_{d,j} = q_e \cdot v_{de,j} + q_d \cdot v_{dd,j}, j = \overline{0, k-1};$
- $t_j \leq t + \alpha_{e,j} \cdot q_e + \alpha_{d,j} \cdot q_d, \alpha_{e,j}, \alpha_{d,j} \in \mathbb{R}^+, j = \overline{0, k-1}.$

Proof. To prove this Lemma, hybrid argument could be applied, because all block cipher keys are independently chosen. We consider a sequence of experiments. The

experiment with number j is exactly the previous one, except for using truly random permutation π_j (and its inverse) instead of E_{K_j} (and its inverse), $j = 0, k-1$. We denote tweakable encryption scheme in j -th experiment by WBE_j .

Consider an adversary $\mathcal{A} \in \mathcal{A}(t, q_e, q_d)$. The following equality holds:

$$\begin{aligned} \text{Adv}_{\text{WBE}[\mathcal{E}]}^{\text{RND-fdeCCA-sector}}(\mathcal{A}) &= \Pr[\mathcal{A}^{\text{WBE}[\mathcal{E}]} \Rightarrow 1] - \Pr[\mathcal{A}^{(\$,\$)} \Rightarrow 1] \\ &= \Pr[\mathcal{A}^{\text{WBE}[\mathcal{E}]} \Rightarrow 1] - \sum_{j=0}^{k-1} \Pr[\mathcal{A}^{\text{WBE}_j} \Rightarrow 1] \quad (25) \\ &\quad + \sum_{j=0}^{k-1} \Pr[\mathcal{A}^{\text{WBE}_j} \Rightarrow 1] - \Pr[\mathcal{A}^{(\$,\$)} \Rightarrow 1] \\ &= \sum_{j=0}^{k-1} \text{Adv}_{\text{WBE}_{j-1}, \text{WBE}_j}^{\text{IND}}(\mathcal{A}) + \text{Adv}_{\text{WBE}_{k-1}, (\$, \$)}^{\text{IND}}(\mathcal{A}), \end{aligned}$$

where $\text{WBE}_{-1} = \text{WBE}[\mathcal{E}]$.

The rest of the proof is straightforward. Let's consider distinguishing advantage $\text{Adv}_{\text{WBE}_{j-1}, \text{WBE}_j}^{\text{IND}}(\mathcal{A})$, $j = 0, k-1$. Consider an adversary $\mathcal{B}_j$ such that:

$$\text{Adv}_{\text{WBE}_{j-1}, \text{WBE}_j}^{\text{IND}}(\mathcal{A}) \leq \text{Adv}_{\mathcal{E}}^{\pm \text{PRP}}(\mathcal{B}_j). \quad (26)$$

The adversary $\mathcal{B}_j$ wraps $\mathcal{A}$ as follows. Before handling queries it samples $k-i+1$ block cipher keys (all keys are sampled uniformly and independently). These keys have indices from $i+1$ to $k-1$. On each encryption (decryption) query $\mathcal{B}_j$ performs encryption (decryption) in the WBE mode. Whenever block encryption (decryption) on i -th key is needed, the following cases could occur:

- $i < j$: the adversary $\mathcal{B}_j$ invokes π_i (π_i^{-1}) to encrypt (decrypt) the block. In this case lazy sampling technique is applied, i.e. the adversary $\mathcal{B}_j$ maintains an internal state;
- $i = j$: the adversary $\mathcal{B}_j$ forwards the block to its own encryption (decryption) oracle;
- $i > j$: the adversary $\mathcal{B}_j$ invokes E_{K_i} ($E_{K_i}^{-1}$) to encrypt (decrypt) the block.

Then the adversary $\mathcal{B}_j$ sends its response (composed according the WBE mode) back to the adversary $\mathcal{A}$. After handling all queries from $\mathcal{A}$, the adversary $\mathcal{B}_j$ returns whatever $\mathcal{A}$ returns. By definition, $\mathcal{B}_j$ perfectly simulates the WBE_j (WBE_{j-1}) scheme for $\mathcal{A}$, if it interacts with random permutation (real block cipher). Therefore, the inequality (26) holds.

To estimate the adversary $\mathcal{B}_j$ resources, we recall how the j -th encryption key is used in encryption and decryption functions of the WBE mode. The adversary $\mathcal{B}_j$ performs $q_{e,j} = q_e \cdot v_{ee,j} + q_d \cdot v_{ed,j}$ encryption queries and $q_{d,j} = q_e \cdot v_{de,j} + q_d \cdot v_{dd,j}$. On each query, the adversary $\mathcal{B}_j$ consumes time for its own oracles queries. This leads to t_j estimation from the Lemma statement. The following inequality holds:

$$\text{Adv}_{\mathcal{E}}^{\pm \text{PRP}}(\mathcal{B}_j) \leq \text{Adv}_{\mathcal{E}}^{\pm \text{PRP}}(t_j, q_{e,j}, q_{d,j}). \quad (27)$$

Next, we need to upper bound the last term in (25). Recall that by definition $\text{WBE}_{k-1} = \text{WBE}[\pi_0, \dots, \pi_{k-1}]$. Since, $\mathcal{A}$ performs in the RND-fdeCCA-sector setting, the following equality holds:

$$\text{Adv}_{\text{WBE}_{k-1}, (\$, \$)}^{\text{IND}}(\mathcal{A}) = \text{Adv}_{\text{WBE}[\pi_0, \dots, \pi_{k-1}]}^{\text{RND-fdeCCA-sector}}(\mathcal{A}).$$

Hence, with the fact that $\mathcal{A}(t, q_e, q_d) \subseteq \mathcal{A}(\infty, q_e, q_d)$ in hand, we obtain:

$$\text{Adv}_{\text{WBE}_{k-1}, (\$, \$)}^{\text{IND}}(\mathcal{A}) \leq \text{Adv}_{\text{WBE}[\pi_0, \dots, \pi_{k-1}]}^{\text{RND-fdeCCA-sector}}(\infty, q_e, q_d). \quad (28)$$

From (25) – (28), we obtain the Lemma statement. **Q.E.D.**

The first summand in (24) requires clarification. It characterizes the WBE mode properties and is considered in *information-theoretic* setting. That is, the adversary is assumed to have unlimited computation resources. Often this value is denoted by $\text{Adv}_{\text{WBE}[\pi_0, \dots, \pi_{k-1}]}^{\text{RND-fdeCCA-sector}}(q_e, q_d)$ or simply by $\text{Adv}_{\text{WBE}[\pi_0, \dots, \pi_{k-1}]}^{\text{RND-fdeCCA-sector}}(q)$, considering $q = q_e + q_d$.

VI. SCHEME INSTANTIATION EXAMPLE

In this section, we discuss what algorithms could be used to instantiate the CS_{SAN} scheme.

With Theorems and Lemmas from previous sections in hand we could reduce security of the CS_{SAN} scheme to the basic problem complexity and security of the scheme components. This reduction is schematically depicted in Figure 11.

Underlying algorithms choice. As defined in the Section 3, the CS_{SAN} scheme consists of several components. This fact is highlighted by writing the scheme as $\text{CS}_{\text{SAN}}[\mathcal{E}, \text{MAC}, \text{KDF}, p]$, where $\mathcal{E} = \text{WBE}[\mathcal{E}]$ is a tweakable encryption scheme based on a block cipher $\mathcal{E}$ and a wide block encryption mode WBE , MAC is message authentication code scheme, KDF is a key derivation function, and p is a partitioning threshold parameter. The $\mathcal{E}$ and MAC schemes form a DAE scheme denoted by AE_{SAN} as in the Section 3.

To properly choose underlying algorithms, one should refer to Theorems 1, 2, and 3. These Theorems establish relations between the CS_{SAN} scheme security and security of its individual components. These requirements are listed below:

- the AE_{SAN} scheme should be secure in the RND-adeCCA-sector setting (Theorems 1 and 2). Since this scheme is built using EtM^* composition method, we require:
 - the $\mathcal{E}$ scheme should be secure in the RND-fdeCPA-sector setting (Theorem 3);
 - the MAC scheme should be secure in the SUF-CMA and the PRF settings (Theorem 3);
- for $p > 0$ case, the KDF function should be secure in the VO-PRF setting (Theorem 2).

Security of the $\mathcal{E} = \text{WBE}[\mathcal{E}]$ scheme could be reduced further to basic problem complexity using Lemma 4.

Given Corollary 1, RND-fdeCPA-sector security could be replaced with (stronger) RND-fdeCCA-sector security.

With these requirements in hand, we give the following example of the CS_{SAN} scheme:

- the XEH mode of operation [11] and Kuznyechik block cipher [14] could be used as the $\tilde{\mathcal{E}}$ scheme;
- the OMAC1 scheme [33, 34] could be used as the MAC scheme;
- the algorithm defined in [35] could be used as the KDF function.

Here we should emphasize that reduction using Lemma 4 is correct in case of the XEH mode and Kuznyechik block cipher. Indeed, the XEH mode uses two independent block cipher keys. Therefore, one could apply Lemma 4 with $k = 2$, $v_{ee,0} = n + 1$, $v_{de,0} = 0$, $v_{ed,0} = 1$, $v_{dd,0} = n$ (the first key is used once in for subkeys derivation and n times to encrypt blocks of data), and $v_{ee,1} = v_{ed,1} = 2$, $v_{de,1} = v_{dd,1} = 0$ (the second key is used twice for subkeys derivation).

Next, we mention several algorithms that could not be

Figure 11. Reduction of the CS_{SAN} security to security of its components and basic problem complexity assuming $\tilde{\mathcal{E}} = WBE[\mathcal{E}]$. Reductions are shown by solid black arrows. Dotted gray arrows show which security model is used in corresponding reduction. Labels near dotted gray arrows refer to corresponding Theorem/Lemma/Corollary.

used in the CS_{SAN} scheme. For instance, there exists a distinguishing attack on the XTS mode in RND-fdeCCA-sector model [7, 36]. Further, the XE (Xor-Encrypt) and XEX (Xor-Encrypt-Xor) modes of operation [37] are secure only in narrow block tweakable PRP setting. One could mount similar attack on these modes as on the XTS mode in [7, 36].

Implementation for the iSCSI protocol. This subsection is dedicated to map abstract notation of the CS_{SAN} scheme on the iSCSI protocol elements.

This mapping is simple. First, by storage object in iSCSI-based SAN we mean a logical unit. Therefore, object number ON is exactly a logical unit number (LUN). Each logical unit is divided into several logical blocks. That is, by storage unit in iSCSI-based SAN we mean a logical block. Logical block address (LBA) corresponds to unit number UN .

The CS_{SAN} .Read algorithm performs $READ(6)$, $READ(10)$, $READ(12)$, $READ(16)$, or $READ(32)$ commands after selecting a proper logical unit. Note, that LUN management (selecting a logical unit) is an external operation preceding performing SCSI commands. The $READ(*)$ commands require the LBA to be specified.

The CS_{SAN} .Write algorithm invokes one of $WRITE(6)$, $WRITE(10)$, $WRITE(12)$, $WRITE(16)$, or $WRITE(32)$ commands. The logical unit is selected before performing these commands. The commands include the LBA only.

The important thing is authentication tags placement. For the iSCSI protocol two common approaches are known: T10 PI approach and metadata sectors.

In the T10 PI approach, a special 8-byte field is associated with each LBA. First 2 bytes store CRC checksum, next 2 bytes store arbitrary logical block application guard, and the last 4 bytes store lower 32 bits of LBA. Without modifying this field layout, one could not store authentication tags there. Even 32-bit tag does not offer necessary security level. Due to this fact, T10 PI could not be used as a storage for tags.

The second approach (metadata sectors) is more usable in our context. On configuration stage several sectors are reserved to store tags. This approach slightly reduces storage space, but does not affect performance dramatically, if tags are stored on the same physical device. Otherwise (if data

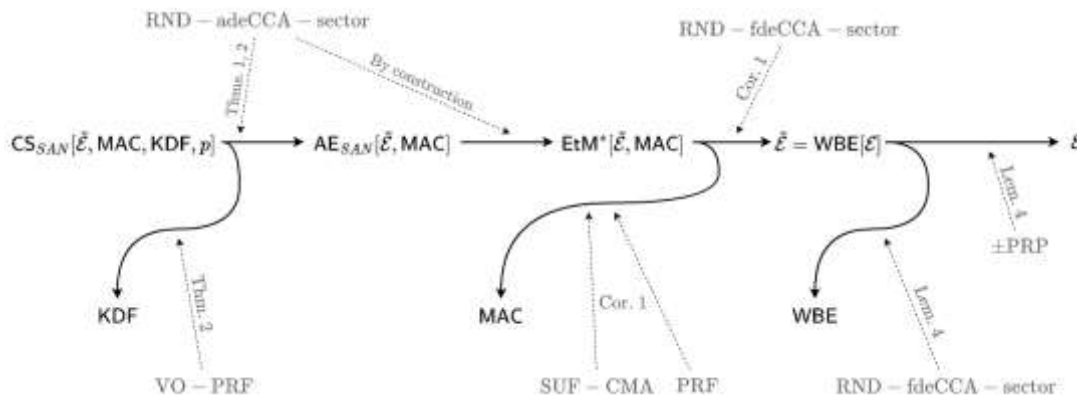
and authentication tags are stored on distinct devices), different access rate and communication overhead could affect the whole scheme performance.

There are several other approaches for storage tags placement. For instance, one could store authentication tags in special filesystem files. However, the CS_{SAN} scheme operates on lower level of I/O stack, than filesystem is located on, and therefore could not directly use filesystem primitives. Another way is to use low-level MPA CRC fields (e.g. while encapsulating iSCSI into iWARP over TCP), but MPA FPDU assumes 32-bit CRC value. Hence, we have the same security issues like with the T10 PI approach due to small tag size.

VII. KEY LIFETIME AND PURPOSE OF $p > 0$ CASE

In this section, we discuss the purpose of the p parameter in the scheme. This parameter is required for key lifetime prolongation.

First, we discuss limits of the $p = 0$ case. The provable security bounds could be used to estimate a lower bound on the amount of data that could be safely processed without key change. To establish an upper bound, algorithmic-combinatorial approach is used [38].



However, these approaches in general case do not consider practical limitations of the attacker. For specific cases this could lead to lowering the key lifetime. Real-life storage devices have their own lifetime due to physical structure. For instance, whenever writing on a solid-state drive (SSD) occurs, corresponding memory cells (NAND flash cells) are slightly damaged due to tunnel injection. Therefore, there exists a limit of how many times data could be written to each sector. Consider an example from [39]. If an SSD has 980 GB capacity and 480 TB total bytes written (TBW) specification, then each sector could be written 480 TB/980 GB = 500 times. In [40], Bogdanov and Nozdrunov give another example with 1 TB capacity and 1200 TB TBW (this case results in 1200 writes per sector in total). This fact establishes an upper bound on total number of adversary queries and could prolongate key lifetime given proper implementation of a protection system.

However, in case of hard disk drive usage, key lifetime prolongation based on physical limitations of queries number is not possible. In this case we provide external key management via the scheme parameter p .

Recall that the CS_{SAN} scheme contains exactly 2^p independent subsystems associated with different sets of storage units. If $p > 0$, then there are more than one such subsystem. Theorem 2 states that adversary advantage against the CS_{SAN} scheme is upper bounded by sum of advantages against these separate systems. Security bound from Theorem 2 encapsulates a variety of possible scenarios.

In the worst case (when UN is fully controlled by the adversary), the adversary could perform all q queries with numbers UN that have the same p most significant bits, i.e., only a single subsystem is involved. This scenario could occur in case of targeted attacks, but implies strong requirements on the adversary capabilities. Security of the whole scheme is reduced to security of the subsystem and the KDF function. In this case, external rekeying [41] should be applied.

In real world, not only worst case should be taken into account. Indeed, cryptographic systems are designed to protect against some specific adversary classes (classification is made by capabilities of the adversaries). Consider more relaxed (and realistic) scenario, when queries are distributed among many storage units. For instance, the adversary forces a legitimate user to download some files or to edit existing documents in some peculiar way. Even more realistic is passive eavesdropping case. A legitimate user performs I/O operations with many units involved. Mathematically speaking, in these relaxed cases, there are $0 \leq q_j \leq q$ queries to j -th subsystem, $j = \overline{0, 2^p - 1}$.

The multinomial theorem [42] gives us necessary conditions for possible key lifetime prolongation in relaxed scenarios. For simplicity, we assume the term related to VO-PRF security of the KDF function in (5) to be negligible. Practically applicable values of p (and corresponding number 2^p of KDF invocations) make our assumption plausible. Fix numbers of encryption q_e and decryption q_m queries. The sum in (5) is strictly less than the right-hand side of inequality in (2), if advantage against AE_{SAN} scheme in RND-adeCCA-sector model is at least quadratic in

number of encryption and/or decryption queries. This could be summarized in the following statement.

Statement 2. Let KDF , $AE_{SAN} = AE_{SAN}[\mathcal{E}, MAC]$, and p be such that:

- $\text{Adv}_{KDF}^{\text{VO-PRF}}(t, 2^p)$ is negligible;
- $\text{Adv}_{AE_{SAN}}^{\text{RND-adeCCA-sector}}(t, q_e, q_d) \leq w(q_e, q_d)$,

where w is a polynomial of degree 2 with respect to its arguments individually without free term, i.e. $w(q_e, q_d) = \alpha q_e^2 + \beta q_d^2 + \gamma(q_e, q_d)$, where γ has degree no more than 1 in q_e and q_d . If the adversary makes at least two queries with different tweaks (i.e. interacts with at least two subsystems), then the bound (5) is strictly less than the bound (2).

Proof. Fix total number of encryption q_e and decryption q_d queries. Let $\theta \leq 2^p$ be the total number of subsystems that the adversary interacts with, and let $q_{e,j}$, $q_{d,j}$ be the number of encryption and decryption queries to the j -th subsystem respectively, $j = \overline{0, \theta - 1}$. Obviously $\sum_{j=0}^{\theta-1} q_{e,j} = q_e$ and $\sum_{j=0}^{\theta-1} q_{d,j} = q_d$. Our goal is to compare security bounds (2) and (5) with assumption $\theta > 1$.

These bounds could be trivially obtained given current Statement assumptions:

$$\text{Adv}_{CS_{SAN}[\mathcal{E}, MAC, KDF, 0]}^{\text{RND-adeCCA-sector}}(t, q_e, q_m) \leq w(q_e, q_d),$$

$$\text{Adv}_{CS_{SAN}[\mathcal{E}, MAC, KDF, p]}^{\text{RND-adeCCA-sector}}(t, q_e, q_m) \leq \sum_{j=0}^{\theta-1} w(q_{e,j}, q_{d,j}).$$

By multinomial theorem [42] we could conclude that given $\theta > 1$ the following inequality holds:

$$\sum_{j=0}^{\theta-1} w(q_{e,j}, q_{d,j}) < w(q_e, q_d).$$

This completes the Statement proof. **Q.E.D.**

Most of existing algorithms that satisfy requirements from the Section 5 section have quadratic upper bound on the adversary advantage, e.g. in [11, 33, 34]. By Theorems and Lemmas from the Section 5, the adversary advantage against the AE_{SAN} scheme instantiated using with algorithms satisfies the condition from the previous paragraph.

VIII. CONCLUSION

In this paper, we proposed a cryptographic scheme for storage area networks and block-oriented storage devices protection. The scheme provides confidentiality and unit-level integrity of the stored data. That is, an adversary could not modify ciphertexts that are stored on the storage device or move them into another location (another storage unit or object). However, to protect against replay attacks or to ensure integrity of the whole storage, additional algorithms should be applied.

The proposed scheme is based on modified Encrypt-then-MAC construction EtM* designed to combine tweakable encryption scheme and message authentication code scheme into a deterministic authenticated encryption scheme. In the present paper, we provided reduction of this construction security to underlying schemes security.

Furthermore, we proven a Lemma that binds tweakable encryption scheme security and underlying block cipher security. This Lemma is crucial for studying security of block cipher-based tweakable encryption schemes built using special block cipher modes of operation. With this Lemma we completed reduction of the proposed scheme security to complexity of basic problem. That is, security of the scheme is reduced down to indistinguishability of a block cipher from a truly random permutation family.

The proposed scheme assumes two master keys shared between devices that should be able to access the data storage. Security proof of the proposed scheme with different key distribution methods is an interesting topic of further research. Another topic is an extension of the scheme to multiple keys.

REFERENCES

- [1] Chukry, S., & Sbeyti, H. (2019). Security Enhancement in Storage Area Network. In *2019 7th International Symposium on Digital Forensics and Security (ISDFS)* (pp. 1–5). IEEE.
- [2] Mahalingam, P., Jayaprakash, N., & Karthikeyan, S. (2009). Enhanced Data Security Framework for Storage Area Networks. In *2009 Second International Conference on Environmental and Computer Science* (pp. 105–110). IEEE.
- [3] Kim, Y., Maino, F., Narasimha, M., & Tsudik, G. (2002). Secure group services for storage area networks. In *First International IEEE Security in Storage Workshop, 2002. Proceedings.* (pp. 80–93). IEEE Comput. Soc.
- [4] Wang, Y., Kumar, A., & Ha, Y. (2014). FPGA-based high throughput XTS-AES encryption/decryption for storage area network. In *2014 International Conference on Field-Programmable Technology (FPT)* (pp. 268–271). IEEE.
- [5] Isobe, T., & Minematsu, K. (2020). Plaintext Recovery Attacks Against XTS Beyond Collisions. In *Selected Areas in Cryptography – SAC 2019* (pp. 103–123). Springer International Publishing.
- [6] Firsov, G., & Koreneva, A. (2022). On One Block Cipher Mode of Operation Used to Protect Data on Block-Oriented Storage Devices. *Modern Information Technologies and IT-Education*, 18(3), 691–701.
- [7] Koreneva, A., & Firsov, G. (2025). O svoystvakh bezopasnosti odnogo rezhima raboty blochnykh shifrov, prednaznachennogo dlya zashchity informacii na nositelyah s blochno-orientirovannoj strukturoj [On security properties of one block ciphers mode of operation for information protection on block-oriented storage devices]. In *Information security systems and tools* (pp. 179–184). Penza state University.
- [8] Kim, Y., Maino, F., Narasimha, M., Kyung Hyune Rhee, & Tsudik, G. (2003). Storage area networking – Secure group key management for storage area networks. *IEEE Communications Magazine*, 41(8), 92–99.
- [9] Kim, Y., Perrig, A., & Tsudik, G. (2000). Simple and fault-tolerant key agreement for dynamic collaborative groups. In *Proceedings of the 7th ACM Conference on Computer and Communications Security* (pp. 235–244). Association for Computing Machinery.
- [10] Minakov, S., Karpov, I., Tikhov, S., & Martynov, I. (2025). Variant postroeniya programmnogo resheniya s gibridnoj kriptograficheskoy sistemoy zashchity dannykh, hranyashchihysya na oblachnom nakopitele, i perspektivnymi rezhimami raboty blochnykh shifrov [A variant for constructing a software solution with a hybrid cryptographic system for protecting data stored on a cloud drive with perspective block ciphers modes of operation]. In *Information security systems and tools* (pp. 179–184). Penza state University.
- [11] Firsov, G., & Koreneva, A. (2024). On improved security bounds of one block ciphers mode of operation for protection of block-oriented system storage devices. *Journal of Computer Virology and Hacking Techniques*, 20(3), 513–523.
- [12] Firsov, G., & Koreneva, A. (2025). Correction to: On improved security bounds of one block ciphers mode of operation for protection of block-oriented system storage devices. *Journal of Computer Virology and Hacking Techniques*, 21(1).
- [13] Ahmetzyanova, L., Alekseev, E., Sedov, G., Smyshlyaeva, E., & Smyshlyaev, S. (2019). Practical significance of security bounds for standardized internally re-keyed block cipher modes. *Mathematical Aspects of Cryptography*, 10(2), 31–46.
- [14] GOST 34.12-2018. Information technology. Cryptographic protection of information. Block ciphers. (2018). Russian National Bureau of Standards.
- [15] Halevi, S., & Rogaway, P. (2003). A Tweakable Enciphering Mode. *Advances in Cryptology - CRYPTO 2003*, 482–499.
- [16] Sarkar, P. (2009). Efficient Tweakable Enciphering Schemes From (Block-Wise) Universal Hash Functions. *IEEE Transactions on Information Theory*, 55(10), 4749–4760.
- [17] Vorobjev, V.I., Rizhkov, C.R., & Phatkieva, R.R. (2015). Zashchita perimetra oblachnykh vychislenij [Cloud computing perimeter protection]. *Programmnye sistemy: teoriya i prilozheniya*, 6(1), 61 – 71.
- [18] Bekker, M.Ya., Terentjev, A.O., Gatchin, Yu.A., & Karmanovskiy, N.S. (2011). Ispol'zovanie cifrovyykh sertifikatov i protokolov SSL/TLS dlya shifrovaniya dannykh pri oblachnykh vychisleniyah [Use of digital certificates and SSL/TLS protocols for data encryption in cloud computing]. *Nauchno-tehnicheskij vestnik informacionnykh tekhnologij, mekhaniki i optiki*, 74(4), 125–130.
- [19] Gunter, E.S., Narutta, N.N., & Shakhov, V.G. (2013). ``Oblachnye'' vychisleniya i problemy ih bezopasnosti [Cloud computations and their security]. *Omskij nauchnyj vestnik*, 120(2), 278–282.
- [20] Minakov, S. (2020). The Main Cryptographic Mechanisms for Protection of Data, Transmitted to Cloud Services and Storage Area Networks. *Cybersecurity issues*, 3(37), 66–75.
- [21] Chakraborty, D., López, C., & Sarkar, P. (2017). Disk encryption: do we need to preserve length? *Journal of Cryptographic Engineering*, 8(1), 49–69.
- [22] Rogaway, P., & Shrimpton, T. (2007). Deterministic authenticated-encryption.
- [23] Fomichev, V., & Koreneva, A. (2020). Encryption performance and security of certain wide block ciphers. *Journal of Computer Virology and Hacking Techniques*, 16(3), 197–216.
- [24] Rogaway, P., & Zhang, Y. (2018). Simplifying Game-Based Definitions: Indistinguishability up to Correctness and Its Application to Stateful AE. In *Advances in Cryptology – CRYPTO 2018* (pp. 3–32). Springer International Publishing.
- [25] Hoang, V., Krovetz, T., & Rogaway, P. (2015). Robust Authenticated-Encryption AEZ and the Problem That It Solves. In *Advances in Cryptology – EUROCRYPT 2015* (pp. 15–44). Springer Berlin Heidelberg.
- [26] Bellare, M., & Namprempre, C. (2008). Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. *Journal of Cryptology*, 21(4), 469–491.
- [27] Smith, P. (2020). *An Introduction to Formal Logic*. Cambridge University Press.
- [28] Ahmetzyanova, L., Alekseev, E., Babueva, A., Nikiforova, L., & Smyshlyaev, S. (2021). IQRA: Incremental Quadratic Re-keying friendly Authentication scheme. *Mathematical Aspects of Cryptography*, 13(3), 5–35.
- [29] Shrimpton, T. (2004). A Characterization of Authenticated-Encryption as a Form of Chosen-Ciphertext Security.
- [30] Kiryukhin, V. (2024). On the security aspects of protocol CRISP. *Mathematical Aspects of Cryptography*, 15(1), 57–81.
- [31] Khati, L. (2019). *Full Disk Encryption and Beyond*. (Doctoral dissertation, Université Paris PSL).
- [32] Namprempre, C., Rogaway, P., & Shrimpton, T. (2014). Reconsidering Generic Composition. In *Advances in Cryptology – EUROCRYPT 2014* (pp. 257–274). Springer Berlin Heidelberg.
- [33] Iwata, T., & Kurosawa, K. (2003). OMAC: One-Key CBC MAC. In *Fast Software Encryption* (pp. 129–153). Springer Berlin Heidelberg.
- [34] Nandi, M. (2009). Improved security analysis for OMAC as a pseudorandom function. *Journal of Mathematical Cryptology*, 3(2).
- [35] R 1323565.1.022–2018. Information technology. Cryptographic protection of information. Key Derivation Functions. (2018). Russian National Bureau of Standards.
- [36] Firsov, G., & Koreneva, A. (2026). O nekotorykh kriptograficheskikh svoystvakh rezhima raboty blochnykh shifrov XTS [On certain cryptographic properties of the XTS block cipher mode of operation]. *IT Security (Russia)*, 33(1), 93–103.
- [37] Rogaway, P. (2004). Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC. In *Advances in Cryptology - ASIACRYPT 2004* (pp. 16–31). Springer Berlin Heidelberg.

- [38] Lavrikov, I., & Shishkin, V. (2019). How much data may be safely processed on one key in different modes? *Mathematical Aspects of Cryptography*, 10(2), 125–134.
- [39] Gunesing, A., Daemen, J., & Mennink, B. (2020). Deck-Based Wide Block Cipher Modes and an Exposition of the Blinded Keyed Hashing Model. *IACR Transactions on Symmetric Cryptology*, 1–22.
- [40] Bogdanov, D., & Nozdrunov, V. (2022). Some properties of the DEC mode of operation of block ciphers. *Mathematical Aspects of Cryptography*, 13(3), 37–44.
- [41] Abdalla, M., & Bellare, M. (2000). Increasing the Lifetime of a Key: A Comparative Analysis of the Security of Re-keying Techniques. In *Advances in Cryptology — ASIACRYPT 2000* (pp. 546–559). Springer Berlin Heidelberg.
- [42] Sachkov, V. (2004). Introduction to combinatorial methods of discrete mathematics. MCNMO.