On security aspects of CRISP

Vitaly Kiryukhin

LLC «SFB Lab», JSC «InfoTeCS», Moscow, Russia
vitaly.kiryukhin@sfblaboratory.ru

Abstract

Using the provable security approach, we analyze CRISP – a standardized Russian cryptographic protocol that aims to ensure confidentiality, integrity of transmitted messages, as well as protection against replay attacks. The protocol is considered as a specific mode of authenticated encryption with associated data (AEAD). We take into account that one key can be used by many protocol's participants and in different cipher suites. We impose requirements for the set of the cipher suites used in the protocol and show that the existing ones meet them. Estimates of the maximum allowable amount of data processed using a single key are also given.

Keywords: CRISP, provable security, AEAD

1 Introduction

CRISP (CRyptographic Industrial Security Protocol) [4] is a secure data transfer protocol designed for use in industrial systems. The security properties that should be provided by the protocol are confidentiality and integrity (or only integrity) of messages and protection against replay attacks.

Important features of the protocol include the following.

Non-Interactivity. Protocol participants do not establish a session, preshared keys are used. Each message contains all (or almost all) the necessary information for processing. Messages may be received out of order.

Multicasting and shared keys. One message from one sender can be intended for many receivers. All users of the information system can share the same secret key.

Dynamic selection of a cipher suite. For each message, the sender can choose any cipher suite from the available ones. Some of them provide confidentiality and integrity, while others provide only integrity.

In this paper, we analyze the cryptographic properties of the protocol by using the provable security approach [8, 9]. We take into account the declared security properties and the above-mentioned protocol features.

Non-Interactivity and simplicity of CRISP encourage us to consider the protocol as a specific *encryption mode*. The lack of authenticated key exchange (AKE) makes it completely irrelevant to use the Canetti-Krawczyk security models [12, 13]. Formal verification tools, such as AVISPA [11], are also useless here for the same reason. We also emphasize that the presented security proofs (based on the approach of Rogaway and Bellare) gives us not only qualitative, but also (more importantly) *quantitative* characteristics, including the "imperfection" of the used encryption algorithms. On the contrary, verification tools usually assume the unconditional ideality of all primitives, and also give only a qualitative result.

The results are presented as follows. In the section 2, the necessary notations and brief information about the provable security paradigm are presented. The third section describes the protocol.

The fourth section is devoted to the general analysis of the protocol's security. We begin with an informal discussion about the capabilities and goals of the adversary. Next, we introduce requirements for the set of the used cipher suites. We show that protocol can be considered as an authenticated encryption with associated data (AEAD) algorithm and then prove that with suitable cipher suites, the CRISP protocol is secure in the relevant threat model.

Section 5 contains the results of the analysis of the existing cipher suites used in CRISP. The known bounds for the cipher modes when used separately or jointly (as AEAD modes) are presented.

In conclusion, estimates of the key capacity (i.e. permissible amount of data processed with one key) and ways to increase them are given.

2 Notations and definitions

We use the following notations throughout the paper:

n – block size in bits; k – key size in bits; $\tau \leq n$ – tag size in bits;
$\oplus$ – bitwise XOR operation; $||$ – concatenation of binary strings;

V^* – the set of all binary strings of a finite length;

V^n – the set of all n-bit strings;

$V^{\leq L}$ – the set of binary strings of length no more than L bits;

$(V^n)^{\leq l}$ – the set of binary strings of length no more than $l \cdot n$ bits, the length of each string is a multiple of n;

$|X|$ – bit length of binary string X;

$\mathrm{Func}(\mathbf{X}, \mathbf{Y})$ – the set of all mappings from the set $\mathbf{X}$ to the set $\mathbf{Y}$;

$\mathrm{Perm}(\mathbf{X})$ – the set of all permutations on the set $\mathbf{X}$;

$X \xleftarrow{\mathrm{R}} \mathbf{X}$ – uniform and random selection of element X from the set $\mathbf{X}$.

Transformations (including ciphers, cipher modes and protocols) are denoted in Sans Serif: E, CTR, CRISP. If the transformation A uses the transformation B from the set of all possible parametrizations, we denote it by $\mathsf{A}[\mathsf{B}]$. The parameter $[\mathsf{B}]$ is omitted when it is clear based on the context.

The adversary is modeled by an interactive probabilistic algorithm that has access to other algorithms (oracles). We denote by $\mathrm{Adv}^{TM}_{\mathsf{Alg}}(\mathcal{A})$ a quantitative characterization (advantage) of the capabilities of the adversary $\mathcal{A}$ in realizing a certain threat, defined by the model TM, for the cryptographic scheme Alg. The resources of $\mathcal{A}$ are measured in terms of time (t) and query (q) complexities. The size of $\mathcal{A}$ description (its source code) is limited by some small value. The query complexity q is measured in the number of adaptively chosen input/output pairs. We assume that $\mathcal{A}$ always uses exactly q unique queries (with no redundant or repeating queries). The algorithm of an oracle (or several oracles) is fixed in the definition of the threat model TM. The result of computations of $\mathcal{A}$ after interacting with oracles $\mathcal{O}_1$, $\mathcal{O}_2$, ... $\mathcal{O}_w$, $w \in \mathbb{N}$ is some binary value x, which is denoted as $\mathcal{A}^{\mathcal{O}_1,\mathcal{O}_2,\ldots,\mathcal{O}_w} \Rightarrow x$.

The maximum of the advantage among all resource constrained adversaries is denoted by

$$\mathrm{Adv}^{TM}_{\mathsf{Alg}}(t,q) = \max_{\mathcal{A}(t',q'):t'\le t,\ q'\le q,} \mathrm{Adv}^{TM}_{\mathsf{Alg}}(\mathcal{A}).$$

Some threat models, which would be addressed later, imply different types of resources, like the number of queries to different oracles, the length of these queries, etc. The advantage for such models is defined in similar way.

The cryptoalgorithm Alg is informally called secure in the threat model TM (TM-secure) if $\mathrm{Adv}^{TM}_{\mathsf{Alg}}(t,q) < \varepsilon$, where ε is some small value determined by the requirements for the strength of the cryptosystem and the resources t and q are comparable to those available to the adversary in practice.

To demonstrate the practical significance of the obtained results, we sometimes substitute heuristic estimates based on assumptions into derived security bounds. The resulting informal estimates are denoted by symbol "$\lessapprox$" meaning "less or equal if the assumptions are true", a slight loss due to omitting of insignificant addends may also occur.

Definitions of frequently used formal models are presented in Appendix A. Other essential definitions are given in the text.

3 Protocol description

3.1 Packet fields

The message (packet) in CRISP consists of the header, payload (`PayloadData`) and tag (`ICV`). The header consists of five fields: `ExternalKeyIdFlag`, `Version`, `CS`, `KeyId`, `SeqNum`. The sizes of the fields are shown in the table below, the total length of all seven fields does not exceed 2048 bytes.

	Name	Symbol	Length in bits	
1	`ExternalKeyIdFlag`	–	1	Header H
2	`Version`	–	15	
3	`CS`	CS	8	
4	`KeyId`	–	from 8 to 1024	
5	`SeqNum`	SN	48	
6	`PayloadData`	P and C	variable	Payload
7	`ICV`	T	variable	Tag

Table 1: List of CRISP-packet fields

The `ExternalKeyIdFlag` and `KeyId` fields indicate the master key K used to process the message. The length of the `KeyId` field is uniquely determined by the first byte of the field itself.

If the flag is zero (`ExternalKeyIdFlag` = 0), then the key is uniquely determined by the field `KeyId`. Otherwise, external information is used.

The field `Version` is fixed and reserved for possible future modifications.

The field `SeqNum` contains the sequence number SN of the message.

The field `CS` contains the identifier CS of the cipher suite. The latter includes:

– `EncryptionAlg` – the encryption/decryption algorithms Enc/Dec (can be set to NULL, meaning that no encryption is applied);

– `MACAlg` – the message authentication code Mac, which computes the τ–bit (`MACLength`) tag T;

– `DeriveIV` – the algorithm DerIv for generating nonces;

– `DeriveKey` – the algorithm KDF for producing derived keys from the master key, and the subalgorithm DerIvKDF that takes SN as input and produces a bit string that is suitable for use as a KDF parameter, thus making its output dependent on SN.

We also refer to the composition of Enc and Mac as AE (authenticated encryption).

The field `PayloadData` contains plaintext P or ciphertext C, depending on the chosen cipher suite. The tag T is computed for all data in fields 1–6

and is contained in the field `ICV`. The tag length is also defined by the cipher suite.

3.2 Common restrictions

The protocol assumes that the sender and the receiver(s) have the same pre-shared master key K with the identifier K_{ID}. Each sender has its own unique identifier `SourceIdentifier` (we denote them by S_{ID}). In the system there is an injective correspondence of the form $K_{ID} \to (K, S_{ID})$, different K_{ID} can correspond to the same K (multiple senders use the same master key). The receiver determines K_{ID} from the fields `ExternalKeyIdFlag`, `KeyId`, and possibly by some external data.

3.3 Initialization of the sequence number

Before using the specific master key K, the sender sets the initial value of $SN \in [0, 2^{48} - 1]$ in an unspecified way. The sequence number must be increasing (for each message from one sender using one key), which includes overflow protection. For each (K, S_{ID}) the receiver initializes the lower $\underline{SN}$ and the upper $\overline{SN}$ bounds of the window (binary vector) W of received messages, $\underline{SN} = \overline{SN} = 0$. The j-th bit of W is set to one if j-th message was received. The receiver stores only bits of W from $\underline{SN}$-th to $\overline{SN}$-th inclusive. The window size is the predefined constant $1 \le Size \le 256$, $(\overline{SN} - \underline{SN}) \le Size$.

3.4 Sender's algorithm

The sender with some S_{ID} selects the master key K (and corresponding K_{ID}), the plaintext P, and the CS-th cipher suite.

1) The sequence number SN is determined by K_{ID}, the value of SN increases by 1.

2) Derived keys K_{MAC} and (if presented) K_{ENC} are computed

$$(K_{ENC}, K_{MAC}) = \mathsf{KDF}(K, prms),$$

where the specific content of $prms$ is determined by the cipher suite and may include CS, S_{ID}, $\mathsf{DerIvKDF}(SN)$, and other parameters.

3) The header H (fields 1-5) is generated, including SN and CS.

4) If the cipher suite provides encryption, then the ciphertext is computed as $C = \mathsf{Enc}(K_{ENC}, IV, P)$, $IV = \mathsf{DerIv}(SN)$, otherwise, $C = P$ is set.

5) The tag $T = \mathsf{Mac}(K_{MAC}, H||C)$ is computed.

6) The message of the form $H||C||T$ is sent.

3.5 Receiver's algorithm

The receiver parses the received message as $H'||C'||T'$ (possibly modified or forged by an attacker) and processes it according to the following algorithm.

1) If the protocol version or the cipher suite specified in H is not supported, then stop processing.

2) K_{ID} is determined by the `KeyId`, `ExternalKeyIdFlag` fields and possibly by external data. Next, K, S_{ID}, $(\underline{SN}, \overline{SN})$, and W are determined by the value of K_{ID}. If the key K is not found, then stop processing.

3) The validity of the sequence number SN is checked:

– if $SN < \underline{SN}$, then stop processing;

– if SN-th bit of W is equal to one, then stop processing.

4) Derived keys K_{MAC} and (if necessary) K_{ENC} are computed $(K_{ENC}, K_{MAC}) = \mathsf{KDF}(K, prms)$.

5) The tag $T'' = \mathsf{Mac}(K_{MAC}, H'||C')$ is computed. If the received and computed tags are not equal ($T' \neq T''$), then stop processing.

6) The window of received messages is updated:

– if $\overline{SN} < SN$, then set $\overline{SN} = SN$ and $\underline{SN} = \min(SN - Size + 1, 0)$;

– the SN-th bit of W is set to one.

7) If the cipher suite provides encryption, then the result is computed as $P' = \mathsf{Dec}(K_{ENC}, IV, C')$, $IV = \mathsf{DerIv}(SN)$, otherwise, $P' = C'$.

4 General security analysis

Mathematically rigorous proof of the security properties of any cryptoalgorithm is possible only in the formal model that includes the qualitative and quantitative capabilities of the adversary, as well as his goals. The discrepancy between the model and practice is a potential source of threats and attacks (see the well-known example of the inconsistency between the model [14] and the attack [15] on the SSL protocol).

The above considerations motivate: to carefully include in the model the capabilities available in practice; to establish the weakest possible goal(s); to stipulate the limitations of the formal model.

Obviously, the adversary knows everything except the keys. The attacker can adaptively chosen plaintexts P and headers H, including the cipher suite CS, master key identifier K_{ID}, sender identifier S_{ID}, and the sequence number SN, but pairs (S_{ID}, SN) are not repeated (i.e. each sender does not use the same sequence number twice with the same key). The adversary also can

drop, reorder, or modify any number of packets.

The adversary's goals are the follows.

1) To get any information about the plaintext P from the ciphertext C (except the length).

2) To make a forgery (create a new valid packet that has not been formed by any sender before).

3) To make a replay (some receiver recognizes the same packet as valid at least twice).

Next, we list the capabilities of the adversary, which he may potentially possess in reality, but which are *not* included in the formal models: changing the protocol version (it is assumed that there is only one); compromising protocol participants (i.e. leakage of the participants' keys); side-channel attacks; fault attacks. The security properties of the protocol when disclosing some keys are shortly discussed at the end of the section.

The non-interactivity of the protocol, along with the aforementioned features and the first two goals of the adversary, prompts us to consider CRISP in the well-established NAE model (*Nonce-based Authenticated Encryption*), see, for example, [26]. This model is also similar to $IND\text{-}CCA3$ proposed in [22]. We prove that even stateless version of the protocol ensures confidentiality and integrity (with the caveat that the sender does not repeat the same SN). Storing states that include windows of the received messages and the sequence numbers provides simple protection against replays.

4.1 Requirements for the cipher suites

We define the cipher suite of the CRISP as the tuple of four algorithms

$$\mathsf{CS} = (\mathsf{KDF}, \mathsf{DerIvKDF}, \mathsf{AE}, \mathsf{DerIv}),$$

where AE can be either a composition of Enc and Mac, or only one algorithm Mac, or a dedicated authenticated encryption mode.

Here we briefly outline the requirements sufficient for the security proof.

Let the master key K be used in several cipher suites. All of them must use the same KDF[1]. Different cipher suites can use keys of different lengths, therefore, KDF must be PRF-secure with variable length of the output (VO-PRF). The input of the KDF must include at least the sender ID S_{ID} and the number CS of the cipher suite. Due to this, different users and different cipher suites will have computationally independent keys. Some bits (we denote

[1]Concurrent usage of different KDFs (for example, CMAC-Magma and HMAC-Streebog) in different cipher suites may not immediately lead to efficient attacks, but when trying to prove formally, some poorly understood basic problems arise during the reduction.

them as $\mathsf{DerIvKDF}(SN)$) of the sequence number SN can also be used as the part of the input. We demand the absence of collisions among nonces, for any $SN \neq SN'$, $\mathsf{DerIvKDF}(SN) \neq \mathsf{DerIvKDF}(SN')$ or/and $\mathsf{DerIv}(SN) \neq \mathsf{DerIv}(SN')$.

If the cipher suite is designed to ensure confidentiality and integrity, then AE must be a secure deterministic AEAD scheme (dedicated or combined). All this properties are also formalized in the NAE model. For cipher suites, from which only integrity is expected, we make the same requirements, but in this case the length of the encrypted data is zero. For example, if $\mathsf{AE} = \mathsf{Mac}$, then PRF-security of Mac is sufficient. Nonce-based schemes, such as Carter-Wegman [6] construction in GCM [10] and UMAC [7] are also suitable.

4.2 Protocol in the NAE model

Stateless version of CRISP is considered in scenario "many senders and one receiver have a single pre-shared key" within the following definitions.

Definition. *The deterministic nonce-based authenticated encryption is the pair of the algorithms*

$$\begin{aligned}\mathsf{AE} :&\mathbf{K} \times \mathbf{N} \times \mathbf{A} \times \mathbf{P} \quad \to \mathbf{C} \times \mathbf{T},\\ \mathsf{AE}^{-1} :&\mathbf{K} \times \mathbf{N} \times \mathbf{A} \times \mathbf{C} \times \mathbf{T} \to \mathbf{P} \cup \{\perp\},\end{aligned}$$

where $\mathbf{K}$, $\mathbf{N}$, $\mathbf{A}$, $\mathbf{P}$, $\mathbf{C}$, $\mathbf{T}$ *are sets of keys, nonces, associated data, plaintexts, ciphertexts, tags, respectively. For any* $(C, T) = \mathsf{AE}(K, N, A, P)$, $P = \mathsf{AE}^{-1}(K, N, A, C, T)$ *is true.*

Definition. *The advantage of* $\mathcal{A}$ *in the model* NAE *for* AE *is*

$$\mathrm{Adv}^{NAE}_{\mathsf{AE}}(\mathcal{A}) = \Pr\left(K \xleftarrow{\mathrm{R}} \mathbf{K} : \mathcal{A}^{\mathsf{AE}_K(\cdot,\cdot,\cdot),\mathsf{AE}_K^{-1}(\cdot,\cdot,\cdot,\cdot)} \Rightarrow 1\right) - \Pr\left(\mathcal{A}^{\$(\cdot,\cdot,\cdot),\perp(\cdot,\cdot,\cdot,\cdot)} \Rightarrow 1\right).$$

The oracle $\$$ *receives the query* (N, A, P) *and returns a random binary string of length* $|P|+ext(P)$ *bits. The extension function* $ext(P)$ *calculates the total length of the tag and padding. The oracle* $\perp$ *always returns error symbol "*$\perp$*". The queries from* $\mathcal{A}$ *to the left oracle (*AE *or* $\$$*) does not contain the same* N*.* $\mathcal{A}$ *does not resend to the right oracle (*AE^{-1} *or* $\perp$*) the answers of the left, that is, it does not query* (N, A, C, T)*, where* (C, T) *is the answer of the left oracle to the query* (N, A, P)*.* $\mathcal{A}$ *makes* q *(resp.* ν*) queries to the left (resp. right) oracle of no more than* l n*-bit blocks each.*

Everywhere else, $N \in \mathbf{N}$ is uniquely determined by the associated data $A \in \mathbf{A}$, hence, the set $\mathbf{N}$ is *implicit*. The algorithm AE can be defined on some *subset* of $\mathbf{A} \times \mathbf{P}$ (with similar changes in AE^{-1}), not on the whole $\mathbf{A} \times \mathbf{P}$.

For CRISP we have:

– the set of all master keys $\mathbf{K} = V^k$;

– $\mathbf{T} = V^{\leq \tau_{\max}}$ (all possible values of the field `ICV`);

– $\mathbf{P} = \mathbf{C} = V^{\leq L_P}$ (`PayloadData`);

– $\mathbf{A} \subseteq \mathbf{A}_{\text{ext}} \times \mathbf{H} \times \mathbf{P}$ (external data plus all possible header values plus `PayloadData` field).

Here we consider external data $A_{\text{ext}} \in \mathbf{A}_{\text{ext}}$ as an "imaginary" packet field. The set $\mathbf{H} \subset V^{\leq L_H}$ contains all possible header values. The values of L_H and L_P do not exceed the packet length (excluding the tag length), $\tau_{\max}$ is the maximum length of the tag among all cipher suites.

The associated data $A \in \mathbf{A}$ explicitly contains the entire header $H \in \mathbf{H}$, and hence the sequence number SN, and the cipher suite number CS. `KeyId` and `ExternalKeyIdFlag` from H, and possibly empty external data $A_{\text{ext}} \in \mathbf{A}_{\text{ext}}$ implicitly correspond to the pair (K, S_{ID}). We assume that this mapping is *injective*, hence, changing the external data leads to change of the key or/and S_{ID} [2]. The length of the `KeyId` field can be different, but the length used is uniquely determined by the first byte of the field. Therefore, changing the length does not violate the injectivity of encoding. The pair $(S_{ID}, SN) \in \mathbf{N}$ is considered as a nonce.

If the chosen cipher suite provides only integrity, then the input of CRISP is $((A_{\text{ext}}, H, P), \emptyset)$, the associated data A consists of the external data A_{ext}, the header H, and the payload P. Otherwise, if both confidentiality and integrity are provided, then the input is $((A_{\text{ext}}, H, \emptyset), P)$. This constraints define the subset of $\mathbf{A} \times \mathbf{P}$ on which CRISP operates.

Theorem 1. *The advantage of the adversary in the NAE model attacking the* CRISP *that uses the cipher suites from the set* $\mathbf{CS} = \{\mathsf{CS}_1, ..., \mathsf{CS}_c\}$,

$$\mathsf{CS}_i = (\mathsf{KDF}, \mathsf{AE}_i, \mathsf{DerIvKDF}, \mathsf{DerIv}_i),\ i = 1, ..., c, \quad \textit{is bounded by}$$

$$\mathrm{Adv}^{NAE}_{\mathsf{CRISP}}(t, q, \nu) \leq \mathrm{Adv}^{VO-PRF}_{\mathsf{KDF}}(t', \kappa) + \sum_{j=1}^{\kappa} \mathrm{Adv}^{NAE}_{\mathsf{AE}^{(j)}}(t', q^{(j)}, \nu^{(j)}),$$

$$\textit{where } \kappa \leq q + \nu, \quad \sum_{j=1}^{\kappa} q^{(j)} = q, \quad \sum_{j=1}^{\kappa} \nu^{(j)} = \nu, \quad \mathsf{AE}^{(j)} \in \{\mathsf{AE}_1, ..., \mathsf{AE}_c\}.$$

Provided that:

1) the input of KDF *contains* S_{ID}, CS, $\mathsf{DerIvKDF}(SN)$;

2) for any $SN \neq SN'$: $\mathsf{DerIvKDF}(SN) \neq \mathsf{DerIvKDF}(SN')$ *or/and* $\mathsf{DerIv}_i(SN) \neq \mathsf{DerIv}_i(SN')$, $i = 1, ..., c$.

[2] The assumption is adequate to the practice. The opposite will require more complex definitions, but does not generate any vulnerabilities. In addition, external data is presented in the packet only "virtually".

The idea of the proof is simple. Restrictions on the use of KDF make it easy to "replace" it with an ideal primitive. Due to this, we get κ *independent* cryptosystems. It remains to apply the "hybrid argument" corresponding to the sum in the estimate. The complete proof is presented in Appendix B.

Corollary. *Let the cipher suites in queries to the left (resp. right) oracle belong to the set* $\mathbf{CS}_{\mathcal{S}}$ *(resp.* $\mathbf{CS}_{\mathcal{R}}$*), and* $\mathbf{CS}_{\mathcal{S}} \cap \mathbf{CS}_{\mathcal{R}}$ *are shared, then*

$$\mathrm{Adv}^{NAE}_{\mathsf{CRISP}}(t,q,\nu) \le \mathrm{Adv}^{VO-PRF}_{\mathsf{KDF}}(t',\kappa=\kappa'+\kappa''+\kappa''')+$$
$$+\sum_{j=1}^{\kappa'}\mathrm{Adv}^{NAE}_{\mathsf{AE}^{(j)}}(t',q^{(j)},0)+\sum_{j=\kappa'+1}^{\kappa''}\mathrm{Adv}^{NAE}_{\mathsf{AE}^{(j)}}(t',q^{(j)},\nu^{(j)})+\sum_{j=\kappa''+1}^{\kappa'''}\mathrm{Adv}^{NAE}_{\mathsf{AE}^{(j)}}(t',0,\nu^{(j)}),$$

where for $j=1,...,\kappa'$, $j=\kappa'+1,...,\kappa''$, $j=\kappa''+1,...,\kappa'''$, $\mathsf{CS}^{(j)}$ *belongs to* $\mathbf{CS}_{\mathcal{S}}\backslash\mathbf{CS}_{\mathcal{R}}$, $\mathbf{CS}_{\mathcal{S}}\cap\mathbf{CS}_{\mathcal{R}}$, $\mathbf{CS}_{\mathcal{R}}\backslash\mathbf{CS}_{\mathcal{S}}$, *correspondingly.*

The security of CRISP against privacy attacks is determined by the weakest set of the sender ($\mathbf{CS}_{\mathcal{S}}$). Forgery attacks can achieve the greatest efficiency when a shared set from $\mathbf{CS}_{\mathcal{S}} \cap \mathbf{CS}_{\mathcal{R}}$ is used (by using $q^{(j)}$ packets protected with the same key), or when a "vulnerable" set supported only by the receiver ($\mathbf{CS}_{\mathcal{R}}\backslash\mathbf{CS}_{\mathcal{S}}$) is used. In the latter case, attempts to forge will essentially be carried out "blindly", this corresponds to zero in $\mathrm{Adv}^{NAE}_{\mathsf{AE}^{(j)}}(t',0,\nu^{(j)})$. A well-known example of the mentioned "vulnerability" is the short tag length, and the corresponding only possible attack is a simple guessing.

Recall that the results above describe a case where the participants have only one shared key. The "many keys" scenario can be reduced in a typical way to the analysis of many single-key independent systems using the "hybrid argument". Obtaining non-trivial results in such conditions is the subject of further research. It seems that this is possible when the protocol is used only to protect integrity, and the sender's ID is included in the packet explicitly.

Also note that considering many receivers instead of one does not seem to lead to meaningful changes in given proofs. Each receiver processes incoming messages independently of the others. The package does not contain a field with any receiver identifier, it is assumed that the receiver can be anyone who has a master key. In practice, the number of forgery attempts ν usually increases linearly with the number of users. In other words, the case of "many receivers" does not lead to a new threat model, but to an increase in adversary resources.

4.3 Replay protection

The security of the CRISP protocol to replay attacks is almost obvious. Indeed, each receiver has the window W of received messages. If a message

with a certain sequence number SN is accepted, then this fact is stored in the window. The second time a message with the same number will be rejected regardless of the content. Messages with numbers less than the lower bound $\underline{SN}$ are also rejected without consideration. The formal proof of this (including cumbersome definitions similar to those proposed in [23, 24, 25, 26]), is not so trivial and due to lack of space, we omit it here.

4.4 Security with leakage of keys

Thanks to the PRF-security of KDF, the CRISP protocol continues to provide some security properties in conditions when some keys become known to an attacker. Obviously, when the master key K is leaked, no security properties are preserved.

If there is a leak of one encryption key K_{ENC}, then the confidentiality of q' messages is violated. The maximum value of q' depends on the algorithm DerIvKDF, that is, from the frequency of changing encryption keys.

If the adversary learns one authentication key K_{MAC}, then each receiver recognizes up to q' forged packets as authentic. Note that in any case, the enemy cannot impose even two packages with the same SN, hence each forgery increases the counter by at least one. Consequently, several forgeries will lead to the key change.

If any number of derived keys is leaked, the adversary cannot efficiently determine the value of any other derived key (and even more so the master key). The opposite would mean that KDF is not PRF-secure.

5 Analysis of the existing cipher suites

The existing version of the CRISP specification contains four "paired" cipher suites (see the table below).

CS	Name	Integrity	Confidentiality	Tag length (τ bit)
1	MAGMA-CTR-CMAC	+	+	32
2	MAGMA-NULL-CMAC	+	−	32
3	MAGMA-CTR-CMAC8	+	+	64
4	MAGMA-NULL-CMAC8	+	−	64

All of them use the block cipher "Magma" [1] $\mathsf{E} : V^k \times V^n \to V^n$ with a key length of $k = 256$ bits and a block length of $n = 64$ bits.

According to GOST R 34.13-2015 [2], the counter mode CTR is used for encryption and CMAC ensures the integrity of the messages. The nonce for

the counter mode is the 32 least significant bits of the sequence number

$$IV = \mathsf{DerIv}(SN) = \mathsf{lsb}_{n/2}(SN),\ SN \in V^{48}.$$

The key derivation function $\mathsf{KDF} : V^k \times V^{\leq L} \times \mathbb{N} \to (V^n)^{\leq d}$ is based on several different calls of CMAC

$$\begin{aligned}\Gamma = \mathsf{KDF}(K, X, d) =& \mathsf{CMAC}[\mathsf{E}](K, \mathsf{byte}(1,1)||X||\mathsf{byte}(n \cdot d, 2))||\\ & \mathsf{CMAC}[\mathsf{E}](K, \mathsf{byte}(2,1)||X||\mathsf{byte}(n \cdot d, 2))||\\ & ...\\ & \mathsf{CMAC}[\mathsf{E}](K, \mathsf{byte}(d,1)||X||\mathsf{byte}(n \cdot d, 2)),\end{aligned}$$

$\mathsf{byte}(x, j)$ is the representation of an integer x as a byte string of length j. The derived keys are computed as

$$K_{MAC}||K_{ENC} = \Gamma,\ d = \frac{2 \cdot k}{n} = 8, \text{ with } CS \in \{1, 3\},$$
$$K_{MAC} = \Gamma,\ d = \frac{k}{n} = 4, \quad \text{with } CS \in \{2, 4\}.$$

The input data X for KDF contains, among other things:
– the number CS of the cipher suite;
– the source identifier S_{ID};
– 35 most significant bits of the sequence number $SN \in V^{48}$
$\mathsf{DerIvKDF}(SN) = \mathsf{msb}_{35}(SN)$.

In KDF the input length of CMAC does not exceed 50 bytes (seven n-bit blocks, $l_{KDF} = 7$). Note that due to the dependency of KDF from 35 bits of SN, no more than $2^{48}/2^{35} = 2^{13}$ packets are processed with the same derived key (or key pair).

5.1 Known bounds for the cipher modes

We list the known bounds in relevant threat models for the ciphers modes used in cipher suites 1-4. Recall that q is the number of protected messages (queries to the oracle); l is the maximum length of a single message in n-bit blocks.

The security proof of $\mathsf{CTR}[\mathsf{E}]$ in the $IND\text{-}CPNA$ model (see the definition in Appendix A) is essentially a consequence of the PRP-PRF Switching Lemma [16] due to which [9]:

$$\mathrm{Adv}^{IND\text{-}CPNA}_{\mathsf{CTR}[\mathsf{E}]}(t, q, l) \leq \mathrm{Adv}^{PRP}_{\mathsf{E}}(t', q \cdot l) + \frac{(q \cdot l)^2}{2^{n+1}}\ t' = t + O(ql).$$

For CMAC a number of estimates in the PRF model are known [17, 18, 19, 20], we quote the last of them.

Theorem ([20, Theorem 3.1]). *The advantage of the adversary in the PRF model attacking the cryptoalgorithm* CMAC *is bounded by*[3]

$$\mathrm{Adv}^{PRF}_{\mathsf{CMAC[E]}}(t,q,l) \le \mathrm{Adv}^{PRP}_{\mathsf{E}}(t', q\cdot l+1) + \frac{16\cdot q^2 + q\cdot l^2 + 4\cdot q\cdot l}{2^n} + \epsilon(q,l),$$

where $t' = t + O(q\cdot l)$, $q\cdot(l+1) \le 2^{n-1}$.

PRF-security of CMAC is sufficient for VO-PRF-security of CMAC-based $\mathsf{KDF}(K,X,d)$. In other words, KDF is indistinguishable from random function when the input and the output lengths are not constant. Let the adversary queries to KDF be (X_1,d_1),...,(X_q,d_q), and $(X_i,d_i) \ne (X_j,d_j)$, $1 \le i < j \le q$. It is easy to see that under such conditions all inputs to the underlying CMAC[E] are different, hence

$$\mathrm{Adv}^{VO\text{-}PRF}_{\mathsf{KDF[CMAC[E]]}}(t,q) \le \mathrm{Adv}^{PRF}_{\mathsf{CMAC[E]}}(t', q\cdot d, l_{KDF}=7).$$

5.2 AEAD-security of composition

It is well known that the algorithms of cipher suites 1-4 produce secure authenticated encryption modes.

Lemma 1. *The advantage of the adversary in the NAE model attacking the cryptoalgorithm*

$$\mathsf{CTR\text{-}CMAC} : \mathbf{K}\times\mathbf{A}\times\mathbf{P}\to\mathbf{C}\times\mathbf{T},$$
$$\mathsf{CTR\text{-}CMAC} : (V^k\times V^k)\times V^{\le l\cdot n}\times V^{\le l\cdot n}\to V^{\le l\cdot n}\times V^{\tau},\ \textit{is bounded by}$$

$$\mathrm{Adv}^{NAE}_{\mathsf{CTR\text{-}CMAC}}(t,q,\nu) \le \mathrm{Adv}^{PRF}_{\mathsf{CMAC[E]}}(t', q+\nu, l) + \mathrm{Adv}^{IND-CPNA}_{\mathsf{CTR[E]}}(t', q, l) + \frac{\nu}{2^{\tau}},$$

$t' = t + O((q+\nu)\cdot l)$. *The query from the adversary to the left oracle is (A,P) and $A = H$.*

Lemma 2. *The advantage of the adversary in the NAE model attacking the cryptoalgorithm*

$$\mathsf{NULL\text{-}CMAC} : \mathbf{K}\times\mathbf{A}\times\mathbf{P}\to\mathbf{C}\times\mathbf{T},$$
$$\mathsf{NULL\text{-}CMAC} : V^k\times V^{\le l\cdot n}\times\emptyset\to\emptyset\times V^{\tau},\ \textit{is bounded by}$$

$$\mathrm{Adv}^{NAE}_{\mathsf{NULL\text{-}CMAC}}(t,q,\nu) \le \mathrm{Adv}^{PRF}_{\mathsf{CMAC[E]}}(t', q+\nu, l) + \frac{\nu}{2^{\tau}},\ t' = t + O((q+\nu)\cdot l).$$

The query from the adversary to the left oracle is $(A,\emptyset)$, $A = H||P$.

[3]The value of $\epsilon(q,l)$ has a bulky form and from a practical point of view is approximately zero. So, for compactness, we omit it.

Thus, both AEAD-modes satisfy the requirements of theorem 1.

The proof of the first lemma is presented in Appendix B. The proof of the second is a direct consequence of the first for the case $\mathbf{P} = \mathbf{C} = \emptyset$.

5.3 Heuristic estimates of basic problem complexity

The security of the cipher suites used in the CRISP protocol is reduced to the single basic problem, namely, the indistinguishability of "Magma" from a random permutation.

Although it is practically impossible to give the exact upper bound for the advantage of the adversary in the PRP model for "Magma", the need to provide practical recommendations motivates us to estimate the value of $\mathrm{Adv}^{PRP}_{\mathsf{Magma}}(\mathcal{A})$ heuristically. The plausible approach is to narrow down the set of all possible algorithms $\mathcal{A}$ with resources (t, q) to the set of currently known methods of constructive cryptanalysis. Methods that require more than 2^k operations for precomputations (for filling the initial memory of the algorithm $\mathcal{A}$) are excluded from the consideration.

The block cipher "Magma" is structurally identical to the GOST 28147-89 algorithm and, unlike its predecessor, has a fixed set of s-boxes, said to provide resistance against the differential [27] and linear [28] methods of the cryptanalysis (see also the design rationale of the 2-GOST cipher [32]).

In [31] a simple distinguisher using "symmetric fixed points" was proposed. For a random permutation $\Pi \in \mathrm{Perm}(V^n)$ the probability of the equality $\Pi(x||x) = x||x$ to hold true for arbitrary $x \in V^{n/2}$ is about 2^{-n}, and for "Magma" it is twice as much. Hence, by checking $q \leq 2^{n/2}$ "symmetric points" the distinguishing advantage is about $\approx q \cdot (2 \cdot 2^{-n} - 2^{-n}) \approx q \cdot 2^{-n}$.

The distinguisher can be built using a key recovery algorithm. If the correct key was found, then the distinguisher's response is "1" (interaction with the cipher), otherwise, the result is "0" (interaction with a random permutation Π). For $q > \frac{k}{n} = \frac{256}{64} = 4$, the probability of a false answer after interacting with Π is almost zero. So, in this case we can consider the distinguishing advantage and the probability of key recovery to be equal.

The success probability of the simple key guessing is about $t \cdot 2^{-k}$.

Two special methods of key recovery are described in [29, 30]. Both methods are arranged in a similar way. Consistently and independently of each other, q known plaintext–ciphertext pairs are considered. For each pair with a probability of 2^{-p}, a "rare event" will occur. The probability of at least one event among q pairs is upper bounded by $q \cdot 2^{-p}$. For each pair, assuming that

the event has occured, 2^c operations are performed and the same number of possible keys are constructed, each of which is checked on other pairs. If the event really happened, then the true key necessarily belongs to the set of tested ones.

The total number of keys constructed is $q \cdot 2^c$. The adversary can perform t computational operations (we assume that one operation is enough to encrypt a block), the proportion of tested keys does not exceed $\frac{t}{q \cdot 2^c}$. The probability of recovering the true key can then be estimated as

$$\frac{q}{2^p} \cdot \frac{t}{q \cdot 2^c} = \frac{t}{2^{p+c}}.$$

However, the probability of success cannot be greater than the probability of a rare event ($q \cdot 2^{-p}$). Therefore, we obtain the upper bound as

$$\min\left(\frac{q}{2^p}, \frac{t}{2^{p+c}}\right).$$

Isobe [29] uses the so-called "reflection property" to mount an attack. The probability of "rare event" is $2^{-p} = 2^{-\frac{n}{2}} = 2^{-32}$. For each pair plaintext-ciphertext, $2^c = 2^{192}$ keys are constructed.

In the attack [30] proposed by Dinur, Dunkelman, Shamir, the "fixed point" is used, $2^{-p} = 2^{-n} = 2^{-64}$, $2^c = 2^{128}$.

Thus, the general from of the heuristic estimation is

$$\mathrm{Adv}^{PRP}_{\mathsf{Magma}}(t,q) \lessapprox \max_{t_1+t_2+t_3=t}\left(\frac{t_1}{2^{256}}, \min\left(\frac{q}{2^{32}}, \frac{t_2}{2^{224}}\right), \min\left(\frac{q}{2^{64}}, \frac{t_3}{2^{192}}\right)\right) + \min\left(2^{-32}, \frac{q}{2^{64}}\right).$$

Simplify for $t \ll 2^{192}$ and arbitrary $q < 2^{32}$

$$\mathrm{Adv}^{PRP}_{\mathsf{Magma}}(t,q) \lessapprox \frac{t}{2^{192}} + \frac{q}{2^{64}}.$$

Therefore, the distinguishing advantage can be considered equal to *zero* for most purposes.

Similarly, other methods of cryptanalysis of the "Magma" cipher can be taken into account in the heuristic estimates.

5.4 Estimates of key capacity

Further, by "key capacity" we mean the permissible amount of data processed under a single key until it should be rotated. Here we discuss approaches to its calculation.

In [3], the concepts of "the maximum allowable probability of a single forgery" (π_{mac}) and "the maximum allowable probability of successful application of cryptanalysis" (π_{enc}) are defined. The NAE model includes both

integrity attacks (forgeries) and privacy attacks (for example, "reading without key"), hence, for any used Alg the inequality must hold true

$$\mathrm{Adv}^{NAE}_{\mathsf{Alg}}(t,q,\nu) < \pi = \min(\pi_{\mathrm{enc}},\pi_{\mathrm{mac}}).$$

For illustrative purposes, we choose $\min(\pi_{\mathrm{enc}},\pi_{\mathrm{mac}}) = 2^{-10}$.

For adversary, we assume, though greatly exaggerating one's real capabilities, that his computational resources are equal to $t \approx t' \approx 2^{128}$ operations.

Recall that κ is the number of derived keys, q (resp. $q' = 2^{13}$) is the number of packets protected with one master (resp. derived) key. For simplicity, we also assume that due to some technical protection, the corresponding number of forgery attempts ν (resp. ν') is much less than q (resp. q'). The maximum packet length is $l \leq \frac{2048\cdot 8}{n} = 2^8$ blocks. KDF uses $d \in \{4,8\}$ calls of CMAC per one derived key.

$\mathrm{Adv}^{PRP}_{\mathsf{Magma}}$ with the declared t can be considered equal to *zero*.

Thus, summing up the above and simplifying the estimates to the most significant terms, we get:

$$\begin{aligned}
\varepsilon_{KDF} \leq \mathrm{Adv}^{PRF}_{\mathsf{CMAC}}(t', \kappa\cdot d = 2^{21}\cdot 8, l_{KDF} = 7) &\lessapprox \frac{16\cdot(\kappa\cdot d)^2}{2^n} = 2^{-12},\\
\varepsilon_{CTR} = \mathrm{Adv}^{IND-CPNA}_{\mathsf{CTR}}(t', q'+\nu' \approx 2^{13}, l = 2^8) &\lessapprox \frac{(q'\cdot l)^2}{2^{n+1}} = 2^{-23},\\
\varepsilon_{CMAC} = \mathrm{Adv}^{PRF}_{\mathsf{CMAC}}(t', q'+\nu' \approx 2^{13}, l = 2^8) &\lessapprox \frac{16\cdot(q')^2}{2^n} = 2^{-34}.
\end{aligned}$$

For the first and the third CS, $\varepsilon_{CS} \approx \varepsilon_{CTR} + \varepsilon_{CMAC} \approx \varepsilon_{CTR}$. For the other two ($CS \in \{2,4\}$), $\varepsilon_{CS} \approx \varepsilon_{CMAC}$.

It is not difficult to see that for each cipher suite $\varepsilon_{CS} < \pi$, the same is true for ε_{KDF} when the number of derived keys $\kappa \leq 2^{21}$. In other words, if, as usual, we consider each derived key *separately*, then "the protocol is secure" with the above restriction on κ.

On the other hand, if we consider the whole protocol and all the keys, then the restriction is now imposed on the sum

$$\mathrm{Adv}^{NAE}_{\mathsf{CRISP}}(t,q,\nu) \leq \mathrm{Adv}^{VO-PRF}_{\mathsf{KDF}}(t',\kappa) + \kappa\cdot\mathrm{Adv}^{NAE}_{\mathsf{CS}}(t',q',\nu') = \varepsilon_{KDF} + \kappa\cdot\varepsilon_{CS},$$

where the second term acquires the greatest importance. So, for the first cipher suite, from $(\varepsilon_{KDF} + \kappa\cdot\varepsilon_{CS}) < \pi$ follows $\kappa < 2^{13}$. The latter is a very strict limitation for practice. In addition, if we replace the derived keys with truly random ones, then $(\kappa\cdot\varepsilon_{CS}) < \pi$ and the estimate does *not* change. In other words, KDF and derived keys do not make CRISP worse.

We emphasize that the above is not an "artifact of provable security". A similar result can be obtained from a constructive point of view, in the sense

of: "the probability of a successful attack on *any one* from κ cryptosystems is approximately κ times greater than the similar probability for one pre-chosen cryptosystem". The choice of the first (each key separately) or second (all keys in the entire system) approaches should be made based on the requirements for a specific information system.

It should be noted that there are many ways to increase the key capacity.

Perhaps the most effective is the use of a cipher with a relatively large block size, namely "Kuznyechik" [1], with $n = 128$, the value of κ is greater than the "unreachable" 2^{54}.

The greatest contribution to the final estimate is made by ε_{CTR}, which degrades quadratically with the growth of the number of blocks. Consequently, some improvements can be achieved by using: the internal re-keying as realized in CTR-ACPKM [5]; truncating of the block cipher output to $s < n$ bits (as provided by the standard [2]); double application of CTR.

6 Conclusion

Using the provable security approach [8, 9] to the analysis of the cryptoalgorithms, we formally proved that the CRISP protocol [4] provides confidentiality, integrity and protection against replays.

CRISP was considered as the algorithm of the authenticated encryption with associated data (AEAD) in the relevant threat model.

We presented the list of sufficient requirements for the cipher suites used in CRISP. The main ones are:

1) the cipher suites used with the same master key must have the same PRF-secure key derivation function;

2) the encryption algorithm and the message authentication code algorithm, applied consequently, must form a secure deterministic AEAD-scheme.

The existing cipher suites [4] satisfy all the specified requirements.

The obtained estimates allowed us to form motivated recommendations on the key capacity.

7 Acknowledgements

The author thanks Alexey Urivskiy, Olga Shemyakina, Andrey Rybkin and Mikhail Borodin for useful discussions. The author is grateful to Andrey Shcherbachenko for many valuable suggestions and corrections. Detailed comments and a list of inaccuracies from anonymous reviewers of CTCrypt 2023 considerably improved the article – special thanks!

References

[1] *GOST R 34.12-2015 – National standard of the Russian Federation – Information technology – Cryptographic data security – Block ciphers*, 2015.

[2] *GOST R 34.13-2015 – National standard of the Russian Federation – Information technology – Cryptographic data security – Block ciphers operation modes*, 2015.

[3] *R 1323565.1.005-2017 – Information technology – Cryptographic data security – Acceptable amount of data to be processed without key change for particular block cipher modes of operation GOST R 34.13-2015*, 2017.

[4] *R 1323565.1.029-2019 – Information technology – Cryptographic data security – Secure exchange protocol for industrial systems*, 2020.

[5] *R 1323565.1.017-2018 – Information technology – Cryptographic data security – Cryptographic algorithms accompanying the use of block encryption algorithms*, 2018.

[6] Wegman M., Carter L., "New hash functions and their use in authentication and set equality", *Journal of Computer and System Sciences*, **22** (1981), 265–279.

[7] Black J., Halevi S., Krawczyk H., Krovetz T., Rogaway P., "UMAC: Fast and Secure Message Authentication", CRYPTO '99, Lect. Notes Comput. Sci., **1666**, 1999, 216–233..

[8] Bellare M., Rogaway P., *Introduction to Modern Cryptography*, 2005.

[9] Rogaway P., *Evaluation of Some Blockcipher Modes of Operation*, CRYPTREC, 2011.

[10] McGrew D. A., Viega J., "The Security and Performance of the Galois/Counter Mode (GCM) of Operation", INDOCRYPT 2004, Lect. Notes Comput. Sci., **3348**, 2004, 343–355.

[11] Armando A. et al., "The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications.", CAV 2005, Lect. Notes Comput. Sci., **3576**, 2005, 281–285.

[12] Canetti R., Krawczyk H., "Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels", EUROCRYPT 2001, Lect. Notes Comput. Sci., **2045**, 2001, 453–474.

[13] LaMacchia, B., Lauter, K., Mityagin, A., "Stronger Security of Authenticated Key Exchange", ProvSec 2007, Lect. Notes Comput. Sci., **4784**, 2007, 1–16.

[14] Krawczyk H., "The Order of Encryption and Authentication for Protecting Communications (Or: How Secure is SSL?)", CRYPTO 2001, Lect. Notes Comput. Sci., **2139**, 2001, 310–331.

[15] Canvel B., Hiltgen A., Vaudenay S., Vuagnoux M., "Password interception in a SSL/TLS channel", CRYPTO 2003, Lect. Notes Comput. Sci., **2729**, 2003, 583–599.

[16] Chang D., Nandi M., "A Short Proof of the PRP/PRF Switching Lemma", *Cryptology ePrint Archive, Report 2008/078*, 2008.

[17] Iwata T., Kurosawa K., "OMAC: One-Key CBC MAC", FSE 2003, Lect. Notes Comput. Sci., **2887**, 2003, 129–153.

[18] Iwata T., Kurosawa K., "Stronger Security Bounds for OMAC, TMAC and XCBC", INDOCRYPT 2003, Lect. Notes Comput. Sci., **2904**, 2003, 402–415.

[19] Nandi M., "Improved security analysis for OMAC as a pseudorandom function", *Journal of Mathematical Cryptology*, **3**:2 (2009), 133–148.

[20] Chattopadhyay S., Jha A., Nandi M., "Towards Tight Security Bounds for OMAC, XCBC and TMAC", ASIACRYPT 2022, 2022.

[21] Bellare M., Namprempre C., "Authenticated Encryption: Relations among notions and analysis of the generic composition paradigm", ASIACRYPT 2000, Lect. Notes Comput. Sci., **1976**, 2000, 531–545.

[22] Shrimpron T., "A Characterization of Authenticated-Encryption as a Form of Chosen-Ciphertext Security", *Cryptology ePrint Archive, Report 2004/272*, 2004.

[23] Kohno T., Palacio A., Black J., "Building Secure Cryptographic Transforms, or How to Encrypt and MAC", *Cryptology ePrint Archive, Report 2003/177*, 2003.

[24] Bellare M., Kohno T., Namprempre C., "Breaking and Provably Repairing the SSH Authenticated Encryption Scheme: A Case Study of the Encode-then-Encrypt-and-MAC Paradigm", *ACM Transactions on Information and Systems Security*, **7**:2 (2004), 206–241.

[25] Boyd C., Hale B., Mjølsnes S. F., Stebila D., "From Stateless to Stateful: Generic Authentication and Authenticated Encryption Constructions with Application to TLS", Cryptographers Track at the RSA Conference 2016, 2016, 55–71.

[26] Rogaway P., Zhang Y., "Simplifying Game-Based Definitions Indistinguishability up to Correctness and Its Application to Stateful AE", CRYPTO 2018, Lect. Notes Comput. Sci., **10992**, 2018, 3–32.

[27] Biham, E., Shamir, A., "Differential cryptanalysis of DES-like cryptosystems", *J. Cryptology*, 1991, 3–72.

[28] Matsui M., "Linear cryptanalysis method for DES cipher", EUROCRYPT'93, Lect. Notes Comput. Sci., **765**, 1994, 386–397.

[29] Isobe T., "A Single-Key Attack on the Full GOST Block Cipher", FSE 2011, Lect. Notes Comput. Sci., **6733**, 2011, 290–305.

[30] Dinur I., Dunkelman O., Shamir A., "Improved Attacks on Full GOST", FSE 2012, Lect. Notes Comput. Sci., **7549**, 2012, 9–28.

[31] Kara O., Karakoc F., "Fixed Points of Special Type and Cryptanalysis of Full GOST", CANS 2012, Lect. Notes Comput. Sci., **7712**, 2012, 86–97.

[32] A. A. Dmukh, D. M. Dygin, G. B. Marshalko, "A lightweight-friendly modification of GOST block cipher", *Mat. Vopr. Kriptogr.*, **5**:2 (2014), 47–55.

A Definitions of the formal models

Definition. The advantage of $\mathcal{A}$ in the model PRP (PRP-CPA – indistinguishability from a random permutation under chosen plaintext attack) for the keyed cryptoalgorithm $\mathsf{E} : \mathbf{K} \times \mathbf{X} \to \mathbf{X}$ is

$$\mathrm{Adv}_{\mathsf{E}}^{PRP}(\mathcal{A}) = \Pr\left(K \xleftarrow{\mathrm{R}} \mathbf{K}; \mathcal{A}^{\mathsf{E}_K(\cdot)} \Rightarrow 1\right) - \Pr\left(\Pi \xleftarrow{\mathrm{R}} \mathrm{Perm}(\mathbf{X}); \mathcal{A}^{\Pi(\cdot)} \Rightarrow 1\right),$$

where $\mathbf{K}$, $\mathbf{X}$ are spaces of the keys and blocks respectively.

Definition. The advantage of $\mathcal{A}$ in the model PRF (PRF-CMA – indistinguishability from a random function under chosen message attack) for the keyed cryptoalgorithm $\mathsf{F} : \mathbf{K} \times \mathbf{X} \to \mathbf{Y}$ is

$$\mathrm{Adv}_{\mathsf{F}}^{PRF}(\mathcal{A}) = \Pr\left(K \xleftarrow{\mathrm{R}} \mathbf{K}; \mathcal{A}^{\mathsf{F}_K(\cdot)} \Rightarrow 1\right) - \Pr\left(\mathsf{R} \xleftarrow{\mathrm{R}} \mathrm{Func}(\mathbf{X}, \mathbf{Y}); \mathcal{A}^{\mathsf{R}(\cdot)} \Rightarrow 1\right),$$

where $\mathbf{K}$, $\mathbf{X}$, $\mathbf{Y}$ are spaces of the keys, messages, and outputs respectively.

Definition. The advantage of $\mathcal{A}$ in the model VO-PRF (variable output – indistinguishability from a random function with variable output length) for the keyed cryptoalgorithm $\mathsf{F} : \mathbf{K} \times V^* \times \mathbb{N} \to V^*$ is

$$\begin{aligned}\mathrm{Adv}_{\mathsf{F}}^{VO\text{-}PRF}(\mathcal{A}) = &\Pr\left(K \xleftarrow{\mathrm{R}} \mathbf{K} : \mathcal{A}^{\mathsf{F}_K(\cdot,\cdot)} \Rightarrow 1\right) - \\ &- \Pr\left(\mathsf{R} \xleftarrow{\mathrm{R}} \mathrm{Func}(V^* \times \mathbb{N}, V^*) : \mathcal{A}^{\mathsf{R}(\cdot,\cdot)} \Rightarrow 1\right).\end{aligned}$$

The query from $\mathcal{A}$ to the oracle is $(X, L) \in V^* \times \mathbb{N}$, where X is data and L is the output length in bits.

Definition. The advantage of $\mathcal{A}$ in the distinguishing two cryptosystems S and $\widetilde{\mathsf{S}}$ (with the same interfaces) is

$$\mathrm{Adv}_{\mathsf{S},\widetilde{\mathsf{S}}}^{IND}(\mathcal{A}) = \Pr\left(\mathcal{A}^{\mathsf{S}(\cdot)} \Rightarrow 1\right) - \Pr\left(\mathcal{A}^{\widetilde{\mathsf{S}}(\cdot)} \Rightarrow 1\right).$$

Definition. The advantage of $\mathcal{A}$ in the model IND-$CPNA$ (indistinguishability under chosen plaintext and nonce attack, also denoted as priv) for the encryption mode $\mathsf{EncMode} : V^k \times V^s \times V^{\leq L} \to V^{\leq L}$ is

$$\mathrm{Adv}_{\mathsf{EncMode}}^{IND\text{-}CPNA}(\mathcal{A}) = \Pr\left(K \xleftarrow{\mathrm{R}} V^k : \mathcal{A}^{\mathsf{EncMode}(K,\cdot,\cdot)} \Rightarrow 1\right) - \Pr\left(\mathcal{A}^{\$(\cdot,\cdot)} \Rightarrow 1\right).$$

$\mathsf{EncMode}(K, \cdot, \cdot)$ is the encryption oracle, that receives the query $(N, P) \in V^s \times V^{\leq L}$ and returns the ciphertext $C \in V^{\leq L}$, where $|C| = |P| + ext(P)$. The oracle $\$(\cdot, \cdot)$ receives the query $(N, P) \in V^s \times V^{\leq L}$ and returns a random binary string of length $|P| + ext(P)$ bits. The adversary $\mathcal{A}$ cannot repeat the value N, each value of $N \in V^s$ is unique. The adversary $\mathcal{A}$ makes q queries of no more than l n-bit blocks each, $l \cdot n \leq L$. The extension function $ext(P)$ computes the length of the required padding.

B Proofs

Theorem 1. *The advantage of the adversary in the NAE model attacking the* CRISP *that uses the cipher suites from the set* $\mathbf{CS} = \{\mathsf{CS}_1, ..., \mathsf{CS}_c\}$,

$$\mathsf{CS}_i = (\mathsf{KDF}, \mathsf{AE}_i, \mathsf{DerIvKDF}, \mathsf{DerIv}_i),\ i = 1, ..., c, \quad \textit{is bounded by}$$

$$\mathrm{Adv}^{NAE}_{\mathsf{CRISP}}(t, q, \nu) \le \mathrm{Adv}^{VO-PRF}_{\mathsf{KDF}}(t', \kappa) + \sum_{j=1}^{\kappa} \mathrm{Adv}^{NAE}_{\mathsf{AE}^{(j)}}(t', q^{(j)}, \nu^{(j)}),$$

$$\textit{where } \kappa \le q + \nu, \quad \sum_{j=1}^{\kappa} q^{(j)} = q, \quad \sum_{j=1}^{\kappa} \nu^{(j)} = \nu, \quad \mathsf{AE}^{(j)} \in \{\mathsf{AE}_1, ..., \mathsf{AE}_c\}.$$

Provided that:

1) the input of KDF *contains* S_{ID}, CS, $\mathsf{DerIvKDF}(SN)$*;*

2) for any $SN \neq SN'$*:* $\mathsf{DerIvKDF}(SN) \neq \mathsf{DerIvKDF}(SN')$ *or/and* $\mathsf{DerIv}_i(SN) \neq \mathsf{DerIv}_i(SN')$, $i = 1, ..., c$.

Proof.

Recall that here we consider the protocol with a single pre-shared master key K. Many senders (each with a unique identifier S_{ID}) use K.

The adversary, according to the NAE model, has access to the pair of oracles. The left "encryption" oracle emulates all senders, the right "verification" oracle corresponds to one receiver. The adversary chooses a specific sender by manipulating associated data A, including unprotected "imaginary" external data A_{ext}, and fields `KeyId` and `ExternalKeyIdFlag` in the header H. According to the assumptions described earlier, an arbitrary change in external data or/and these fields entails a change in (K, S_{ID}). Due to the uniqueness of K, this means that the S_{ID} must be changed.

According to the requirements of the theorem, all cipher suites use the same KDF. Consider the CRISP-I protocol, in which a random function $\mathsf{R} \in \mathrm{Func}(V^* \times \mathbb{N}, V^*)$ is used instead of KDF. Let's construct algorithm $\mathcal{B}$ so that the inequality holds

$$\mathrm{Adv}^{IND}_{\mathsf{CRISP},\mathsf{CRISP\text{-}I}}(\mathcal{A}) \le \mathrm{Adv}^{VO\text{-}PRF}_{\mathsf{KDF}}(\mathcal{B}).$$

Each query from $\mathcal{A}$ to either of its two oracles may require the computation of derived keys. Algorithm $\mathcal{B}$ emulates one of two protocols (CRISP or CRISP-I) for $\mathcal{A}$. Hence, $\mathcal{B}$ makes up to $\kappa \le (q + \nu)$ queries to the oracle (KDF or R). One response from the oracle is the derived key $K^{(j)}$ for the cipher mode $\mathsf{AE}^{(j)}$, $j = 1, ..., \kappa$. So, $\mathcal{B}$ has all derived keys and, therefore, can perfectly simulate the protocol. The result of $\mathcal{B}$ is equal to the result of $\mathcal{A}$.

In fact, CRISP-I contains κ subsystems (i.e. some $\mathsf{AE}^{(j)}$ with the key $K^{(j)}$) *independent* of each other. By virtue of condition 1, the following subsystems have independent keys:

– with different cipher suites (CS is the part of KDF input);

– with the same cipher suites, but with the different senders (S_{ID} is also the part of KDF input);

– with the same cipher suites and the same S_{ID}, but with different $\mathsf{DerIvKDF}(SN)$.

Recall that the pair (S_{ID}, SN) is considered as nonce in CRISP and CRISP-I – the sender does not repeat its own sequence numbers. Along with this, different SN can correspond to the same $IV = \mathsf{DerIv}_i(SN)$ for some $i = 1, ..., c$. Due to condition 2, within any of the κ subsystems, the IV values are also not repeated.

The adversary chooses one subsystem for interaction by specifying associated data A (this includes S_{ID}, CS, SN) in the query.

In the j-th system, the adversary can make $q^{(j)}$ (resp. $\nu^{(j)}$) queries to the "encryption" (resp. "verification") oracle. The maximum of $q^{(j)}$ depends on the number of bits in SN that do not affect $\mathsf{DerIvKDF}(SN)$. All forgery attempts can be carried out using a single (S_{ID}, CS, SN), hence, $\nu^{(j)} \leq \nu$. The total numbers of queries are $\sum_{j=1}^{\kappa} q^{(j)} = q$ and $\sum_{j=1}^{\kappa} \nu^{(j)} = \nu$.

Thanks to the independence of the keys in CRISP-I, we can use the so-called "hybrid argument". Let we have the sequence of the protocols[4]

$$\mathsf{CRISP\text{-}I}^{(0)}, ..., \mathsf{CRISP\text{-}I}^{(\kappa)},$$

where $\mathsf{CRISP\text{-}I}^{(0)} = \mathsf{CRISP\text{-}I}$ and $\mathsf{CRISP\text{-}I}^{(\kappa)}$ is the "ideal" (all κ pairs of oracles are $(\$, \perp)$). In $\mathsf{CRISP\text{-}I}^{(j)}$, $0 < j < \kappa$, all pairs of oracles with indexes $1, ..., j$, are replaced by the "ideal" $(\$, \perp)$ ones, all other pairs are "real" $(j+1, ..., \kappa)$.

If $\mathcal{A}$ can effectively distinguish $\mathsf{CRISP\text{-}I}^{(j-1)}$ and $\mathsf{CRISP\text{-}I}^{(j)}$, then there is $\mathcal{B}^{(j)}$ that can effectively attack $\mathsf{AE}^{(j)}$ in the NAE model. Before starting interactions, $\mathcal{B}^{(j)}$ generates keys $K^{(j')}$ for subsystems with indexes $j' > j$. After that, for any query from $\mathcal{A}$, $\mathcal{B}^{(j)}$ determines the index j' of the oracle pairs by the associated data in the query. If $j' < j$, then $\mathcal{B}^{(j)}$ simulates "ideal" oracle ($\$ or $\perp$). If $j' = j$, then $\mathcal{B}^{(j)}$ makes the corresponding query to its own oracle and returns the response to $\mathcal{A}$. In other cases ($j' > j$), $\mathcal{B}^{(j)}$ simulates "real" oracle by using a self-generated key $K^{(j')}$. If $\mathcal{B}^{(j)}$ interacts with the "real" (resp. "ideal") oracles, then $\mathsf{CRISP\text{-}I}^{(j-1)}$ (resp. $\mathsf{CRISP\text{-}I}^{(j)}$) is perfectly simulated for $\mathcal{A}$. The result of $\mathcal{B}^{(j)}$ is equal to the result of $\mathcal{A}$.

[4]Speaking more formally, we can enumerate all possible $\kappa_{\max}$ triples $(S_{ID}, CS, \mathsf{DerIvKDF}(SN))$ and consider all corresponding subsystems. In general, $\kappa_{\max} \geq \kappa$, but the adversary does not make any queries to $(\kappa_{\max} - \kappa)$ systems, and hence, using $\kappa_{\max}$ instead of κ does not affect the result.

The advantage of $\mathcal{A}$ is bounded by

$$\mathrm{Adv}^{IND}_{\mathsf{CRISP\text{-}I}^{(j-1)},\mathsf{CRISP\text{-}I}^{(j)}}(\mathcal{A}) \le \mathrm{Adv}^{NAE}_{\mathsf{AE}^{(j)}}(\mathcal{B}_j),$$

$\mathcal{B}_j$ makes $q^{(j)}$ and $\nu^{(j)}$ queries.

By the triangle inequality we obtain

$$\mathrm{Adv}^{IND}_{\mathsf{CRISP\text{-}I}^{(0)},\mathsf{CRISP\text{-}I}^{(\kappa)}}(\mathcal{A}) \le \sum_{j=1}^{\kappa} \mathrm{Adv}^{NAE}_{\mathsf{AE}^{(j)}}(\mathcal{B}_j),$$

and due to the arbitrariness of the algorithm $\mathcal{A}$ the original statement of the theorem is true.

□

Lemma 1. *The advantage of the adversary in the NAE model attacking the cryptoalgorithm*

$\mathsf{CTR\text{-}CMAC} : \mathbf{K} \times \mathbf{A} \times \mathbf{P} \to \mathbf{C} \times \mathbf{T}$,

$\mathsf{CTR\text{-}CMAC} : (V^k \times V^k) \times V^{\le l\cdot n} \times V^{\le l\cdot n} \to V^{\le l\cdot n} \times V^{\tau}$, *is bounded by*

$$\mathrm{Adv}^{NAE}_{\mathsf{CTR\text{-}CMAC}}(t,q,\nu) \le \mathrm{Adv}^{PRF}_{\mathsf{CMAC[E]}}(t',q+\nu,l) + \mathrm{Adv}^{IND-CPNA}_{\mathsf{CTR[E]}}(t',q,l) + \frac{\nu}{2^{\tau}},$$

$t' = t + O((q+\nu)\cdot l)$. *The query from the adversary to the left oracle is* (A,P) *and* $A = H$.

Proof.

$\mathsf{CTR\text{-}CMAC}$ is a pair of the algorithms denoted here by

$$(\mathsf{AE}[\mathsf{CTR},\mathsf{CMAC}], \mathsf{AE}^{-1}[\mathsf{CTR},\mathsf{CMAC}]).$$

Recall that the nonce IV is determined by the associated data A and is equal to the 32 least significant bits of the sequence number SN.

By definition, the advantage of the adversary $\mathcal{A}$ is

$$\begin{aligned}\mathrm{Adv}^{NAE}_{\mathsf{CTR\text{-}CMAC}}(\mathcal{A}) = \Pr((K_{ENC},K_{MAC}) \xleftarrow{\mathrm{R}} V^k \times V^k;\\ \mathcal{A}^{\mathsf{AE}((K_{ENC},K_{MAC}),\cdot,\cdot),\mathsf{AE}^{-1}((K_{ENC},K_{MAC}),\cdot,\cdot,\cdot)} \Rightarrow 1) -\\ - \Pr(\mathcal{A}^{\$(\cdot,\cdot),\perp(\cdot,\cdot,\cdot)} \Rightarrow 1),\end{aligned}$$

where the oracle \$ returns a random binary string of length $|P| + \tau$ in the response to the query (A,P), and the oracle $\perp$ always returns symbol "$\perp$"

Firstly, CMAC is replaced by a random function $\mathsf{R} : V^{\le l\cdot n} \to V^{\tau}$. Let $\mathcal{A}$ be able to effectively distinguish the original cryptoalgorithm from the modified one, then there is $\mathcal{B}_1$ that can distinguish CMAC from a random

function. The algorithm $\mathcal{B}_1$ generates $K_{ENC} \xleftarrow{R} V^k$ and stores it. The query (A, P) from $\mathcal{A}$ to the left oracle is processed by $\mathcal{B}_1$ as follows: read IV from associated data A; compute $C = \mathsf{CTR}(K_{ENC}, IV, P)$; receive the tag by the query to the oracle $T = \mathcal{O}(A||C)$, $\mathcal{O} \in \{\mathsf{CMAC}, \mathsf{R}\}$; return (C, T) to $\mathcal{A}$. The query (A, C, T) from $\mathcal{A}$ to the oracle AE^{-1} is processed by $\mathcal{B}_1$ as follows: receive the tag $T' = \mathcal{O}(A||C)$; compare $T' = T$; if equality is not satisfied, return character "$\perp$"; otherwise, read IV from A and return $P = \mathsf{CTR}(K_{ENC}, IV, C)$.

$\mathcal{B}_1$ perfectly simulates for $\mathcal{A}$ the cryptoalgorithm $\mathsf{AE}[\mathsf{CTR}, \mathsf{CMAC}]$ or $\mathsf{AE}[\mathsf{CTR}, \mathsf{R}]$ (and, of course, the corresponding AE^{-1}). The result of $\mathcal{B}_1$ is equal to the result of $\mathcal{A}$, hence

$$\begin{aligned}&\Pr((K_{ENC}, K_{MAC}) \xleftarrow{R} V^k \times V^k;\\ &\quad \mathcal{A}^{\mathsf{AE}[\mathsf{CTR},\mathsf{CMAC}]((K_{ENC},K_{MAC}),\cdot,\cdot),\mathsf{AE}^{-1}[\mathsf{CTR},\mathsf{CMAC}]((K_{ENC},K_{MAC}),\cdot,\cdot,\cdot)} \Rightarrow 1) -\\ -&\Pr(K_{ENC} \xleftarrow{R} V^k; \mathsf{R} \xleftarrow{R} \mathrm{Func}(V^{\leq l\cdot n}, V^\tau);\\ &\quad \mathcal{A}^{\mathsf{AE}[\mathsf{CTR},\mathsf{R}](K_{ENC},\cdot,\cdot),\mathsf{AE}^{-1}[\mathsf{CTR},\mathsf{R}](K_{ENC},\cdot,\cdot,\cdot)} \Rightarrow 1) \leq \mathrm{Adv}^{PRF}_{\mathsf{CMAC}[\mathsf{E}]}(\mathcal{B}_1).\end{aligned}$$

The number of queries from $\mathcal{B}_1$ is equal to the number of queries from $\mathcal{A}$ and is $q + \nu$. The number of queries from $\mathcal{B}_1$ is equal to $q + \nu$.

Secondly, the oracle AE^{-1} is replaced by $\perp$. The advantage of the adversary in this case is bounded by the probability of making a forgery in ν attempts. Let AE^{-1} receive the query $(A, C, T) \notin \{(A_1, C_1, T_1), ..., (A_q, C_q, T_q)\}$. If $A||C \notin \{A_1||C_1, ..., A_q||C_q\}$, then the guessing probability is

$$\Pr(\mathsf{R}(A||C) = T) = 2^{-\tau}.$$

If $A||C = A_i||C_i$, $i = 1, ..., q$, then $T \neq T_i$, and hence $\Pr(\mathsf{R}(A||C) = T) = 0$. Therefore, the probability of at least one correct guess in ν attempts is at most

$$\begin{aligned}&\Pr(\mathcal{A}^{\mathsf{AE}[\mathsf{CTR},\mathsf{R}](K_{ENC},\cdot,\cdot),\mathsf{AE}^{-1}[\mathsf{CTR},\mathsf{R}](K_{ENC},\cdot,\cdot,\cdot)} \Rightarrow 1) -\\ -&\Pr(\mathcal{A}^{\mathsf{AE}[\mathsf{CTR},\mathsf{R}](K_{ENC},\cdot,\cdot),\perp(\cdot,\cdot,\cdot)} \Rightarrow 1) \leq \frac{\nu}{2^\tau}.\end{aligned}$$

Thirdly, the oracle $\mathsf{AE}[\mathsf{CTR}, \mathsf{R}]$ is replaced by \$. If replacement can be effectively detected, then there is the algorithm $\mathcal{B}_2$ effectively attacking CTR in the $IND\text{-}CPNA$ model. The query (A, P) from $\mathcal{A}$ to the left oracle is processed by $\mathcal{B}_2$ as follows: read IV from associated data A; get $C = \mathcal{O}(IV, P)$ from the oracle $\mathcal{O} \in \{\mathsf{CTR}, \$\}$; generate $T \xleftarrow{R} V^\tau$; return the response (C, T). Due to the fact that all queries (A, P) are different, the random generation

$T \xleftarrow{\mathrm{R}} V^{\tau}$ corresponds perfectly to the behavior of a random function R. The response to the query from $\mathcal{A}$ to $\perp$ is trivially simulated. The result of $\mathcal{B}_2$ is equal to the result of $\mathcal{A}$, and therefore we obtain,

$$\Pr(\mathcal{A}^{\mathsf{AE}[\mathsf{CTR},\mathsf{R}](K_{ENC},\cdot,\cdot),\perp(\cdot,\cdot,\cdot)} \Rightarrow 1) - \Pr(\mathcal{A}^{\$(\cdot,\cdot),\perp(\cdot,\cdot,\cdot)} \Rightarrow 1) \leq \mathrm{Adv}^{IND-CPNA}_{\mathsf{CTR}[\mathsf{E}]}(\mathcal{B}_2).$$

Algorithm $\mathcal{B}_2$ makes no more than q queries. We get the stated inequality as the sum of the advantages using the triangle inequality. $\square$