Threshold Encryption into Multiple Ciphertexts

Martin Stanek

Department of Computer Science
Comenius University
stanek@dcs.fmph.uniba.sk

Abstract. We propose (T, N) multi-ciphertext scheme for symmetric encryption. The scheme encrypts a message into N distinct ciphertexts. The knowledge of the symmetric key allows decryption of the original message from any ciphertext. Moreover, knowing $T + 1$ ciphertexts allows efficient recovery of the original message without the key (and without revealing the key as well). We define the security property of the scheme, and prove the security of the proposed scheme. We discuss several variants of the basic scheme that provides additional authenticity and efficiency.

1 Introduction

Imagine a situation where a secure backup of data has to be stored on multiple locations. It is easy to achieve this using an encryption – we encrypt the data and store them on as many locations as we want. In order to recover the data again, we need a decryption key and a backup from at least one location. We can reduce the need for "full" key-management by distributing the key via secret sharing scheme (and storing particular shares on specified locations). Then one can recover the data from a single backup (provided (s)he knows the key) or recover the data having backups from multiple locations (provided the key can be reconstructed from the shares). However this approach has a drawback – the key is revealed/compromised by this reconstruction (so it cannot be used for other data/purposes).

Imagine another situation where data are written to multiple systems (disks) for redundancy reasons. Because of confidentiality requirements, each copy of the data must be encrypted. In order to allow fast read operation of the data, we want a fast recovery (i.e. without using a decryption algorithm) of data from encrypted copies as long as we have access to at least some predefined number of copies.

Our contribution. We propose (T, N) multi-ciphertext scheme for symmetric encryption (where $1 \leq T \leq N$). The scheme encrypts a message into N distinct ciphertexts. The knowledge of the symmetric key allows decryption of the original message from any ciphertext. Moreover, knowing $T + 1$ ciphertexts allows efficient recovery the original message without the key (and without revealing the

key as well). A $(1, 2)$ multi-ciphertext scheme called "double ciphertext mode" was proposed in [3]. Our work solves an open problem formulated there and can be viewed as a generalization of their scheme.

We define the security property of the scheme that differs from the definition provided in [3], which is based on indistinguishability of ciphertexts and random binary strings. We follow a "semantic security" approach – indistinguishability of encryptions of real-or-random plaintexts, introduced by Bellare et al. [1].

We prove the security of a concrete implementation of multi-ciphertext scheme (called multiple ciphertext mode), using a proof techniques similar to [1]. We discuss several variants of the basic scheme that provides additional efficiency and authenticity.

The paper is organized as follows. Section 2 introduces basic notions. A multi-ciphertext scheme and its security are defined in Sect. 3. Multiple ciphertext mode is proposed in Sect. 4, together with proofs of its security. Section 5 contains variants of multiple ciphertext mode that address some efficiency issues and authenticity requirements.

2 Preliminaries

We denote $[l] = \{1, 2, \ldots, l\}$ for a positive integer l. Let k and n be positive integers. A block cipher is a function $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$, where for each key $K \in \{0,1\}^k$, the function $E_K(\cdot) = E(K, \cdot)$ is a permutation on $\{0,1\}^n$. Let $\mathrm{Bloc}(k, n)$ be the set of all block ciphers with k-bit keys and n-bit blocks. The inverse of block cipher E is denoted by E^{-1}.

The construction will use the arithmetic in finite field $\mathrm{GF}(2^n)$, i.e. n-bit blocks are sometimes treated as elements in $\mathrm{GF}(2^n)$. For this purpose we assume some representation of $\mathrm{GF}(2^n)$ is fixed and known. We denote the addition in $\mathrm{GF}(2^n)$, i.e. the bitwise XOR operation of two binary vectors as $\oplus$. Let $\mathrm{bin}(a)$ be an element of $\mathrm{GF}(2^n)$ corresponding to the n-bit binary representation of integer $a \in \{0, \ldots, 2^n - 1\}$, e.g. $\mathrm{bin}(3) = x + 1$, $\mathrm{bin}(6) = x^4 + x^2$ etc. A scalar product of vectors $\alpha = (a_1, \ldots, a_l)$, $\beta = (b_1, \ldots, b_l)$, where all $a_i, b_i \in \mathrm{GF}(2^n)$, is denoted as $\langle \alpha, \beta \rangle$, i.e. $\langle \alpha, \beta \rangle = a_1 b_1 \oplus \ldots \oplus a_l b_l$.

The experiment of choosing a random element x from the finite set S will be denoted by $x \xleftarrow{\$} S$. Let $\mathrm{Func}(n, n')$ be the set of all functions from $\{0,1\}^n$ to $\{0,1\}^{n'}$.

An adversary (or sometimes a distinguisher) is a probabilistic algorithm that interacts with one or more oracles (written as superscripts) and outputs 0 or 1.

Let $F : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^{n'}$ be a function with key space $\{0,1\}^k$, domain $\{0,1\}^n$ and range $\{0,1\}^{n'}$. A distinguisher D tries to distinguish F from a random function, having oracle access to one of them. We define its advantage as follows:

$$\mathbf{Adv}^{\mathrm{prf}}_F(D) = \Pr[K \xleftarrow{\$} \{0,1\}^k : \ D^{F_K(\cdot)} \Rightarrow 1] - \Pr[f \xleftarrow{\$} \mathrm{Func}(n, n') : \ D^{f(\cdot)} \Rightarrow 1].$$

A function F is called (t, q, ε)-secure pseudo-random function if for any probabilistic distinguisher D that runs in time t and makes q queries $\mathbf{Adv}^{\mathrm{prf}}_F(D) \leq \varepsilon$.

Similarly, we can define (t, q, ε)-secure pseudo-random permutation; the only distinctions are requirements for $n = n'$, F_K being a permutation on $\{0,1\}^n$ for all $K \in \{0,1\}^k$, and the distinguisher tries to distinguish F from a random permutation. Secure pseudo-random permutation is sufficiently secure pseudo-random function (see [1, 4], the difference is negligible and related to the birthday bound of finding a collision). Therefore, similarly to [1], we use pseudo-random function instead of pseudo-random permutation to model a block cipher in the analysis of our scheme.

3 A Multi-ciphertext Scheme

Let T, N, m, l be positive integers. Let $\mathcal{M} = \{0,1\}^{nm}$ be the message space, and $\mathcal{C} = \{0,1\}^{nm}$ be the ciphertext space. A (T, N) multi-ciphertext scheme is a quadruple of algorithms $(\mathcal{K}, \mathcal{E}, \mathcal{D}, \mathsf{Recover})$:

- $\mathcal{K}$ – the key generation algorithm (we also denote the key space as $\mathcal{K}$)
- $\mathcal{E} : \mathcal{K} \times \mathcal{E} \to \mathcal{C}^N \times \{0,1\}^l$ – the encryption algorithm. It takes a key and a message to produce N ciphertexts and an l-bit tag τ. For an $M \in \mathcal{M}$, $K \in \mathcal{K}$ we denote $\mathcal{E}(K, M) = \mathcal{E}_K(M) = (C_1, \ldots, C_N, \tau)$. The tag τ is used to randomize the encryption and will be generated as a random bit-string in our construction. Moreover, in order to allow a distributed computation of individual ciphertexts, we require that knowing the key and τ allows computation of any C_i. We denote the partial encryption functions as $\mathcal{E}^1, \ldots, \mathcal{E}^N$, i.e. $C_i = \mathcal{E}^i_K(M, \tau)$ for all $i = 1, \ldots, N$.
- $\mathcal{D} : \mathcal{K} \times \mathcal{C} \times [N] \times \{0,1\}^l \to \mathcal{M}$ – the decryption algorithm. It takes a key, a ciphertext, its index and a tag to produce the original message. We require the correctness of the scheme, i.e.

$$\forall K \in \mathcal{K}\ \forall M \in \mathcal{M}\ \forall \tau \in \{0,1\}^l\ \forall i \in [N] : \quad \mathcal{D}_K(\mathcal{E}^i_K(M, \tau), i, \tau) = M.$$

- $\mathsf{Recover}$ – the algorithm for message recovery without using the key. It takes at least $T + 1$ ciphertexts and the tag τ. The algorithm outputs the original message. More formally:

$$\forall K \in \mathcal{K}\ \forall M \in \mathcal{M}\ \forall \tau \in \{0,1\}^l\ \forall I \subseteq [N]\ (|I| > T) : \\ \mathsf{Recover}(\{(i, E^i_K(M, \tau)) \mid i \in I\}, \tau) = M.$$

Remark 1. Whether the tag is generated randomly by the scheme or randomly generated and supplied by application is an unimportant technical detail. In practice we can choose both approaches (however, for our definition we prefer the tag to be generated internally). For other modifications, the tag can be generated by suitable pseudorandom function and serve for authentication purposes (see Sect. 5.3 or [3]).

Security. The security definition of a multi-ciphertext scheme is modeled according real-or-random property proposed by Bellare et al. [1]. An adversary is any probabilistic polynomial time machine with access to an oracle. In the first ("real") scenario the oracle returns encryption of given plaintext. In the second ("random") scenario the oracle returns encryption of randomly chosen string of equal length. The adversary tries to distinguish between these two scenarios.

The multi-ciphertext scheme allows recovery of the original message from $T+1$ or more ciphertexts. Therefore, we have to restrict the number of ciphertexts (at least T) the adversary can obtain from the encryption oracle (the tag is available for the adversary as well). On the other hand, we allow the adversary to choose which T ciphertexts it receives for each query. We denote the selection $\mathcal{E}_K(\cdot) \downharpoonright (I)$, where $I = \{i_1, \ldots, i_T\} \subseteq [N]$ and $|I| = T$, i.e. $\mathcal{E}_K(\cdot) \downharpoonright (I) = (C_{i_1}, \ldots, C_{i_T}, \tau)$.

More formally, let A be an adversary, and we denote the (privacy) advantage of A for a multi-ciphertext scheme Γ:

$$\mathbf{Adv}^{\mathrm{priv}}_{\Gamma}(A) = \Pr\left[K \xleftarrow{\$} \mathcal{K} :\ A^{\mathcal{E}_K(\cdot)\downharpoonright(.)} \Rightarrow 1\right] - \Pr\left[K \xleftarrow{\$} \mathcal{K} :\ A^{\mathcal{E}_K(\$^{|\cdot|})\downharpoonright(.)} \Rightarrow 1\right],$$

where $\$^{|\cdot|}$ on input M generates $M' \xleftarrow{\$} \{0,1\}^{|M|}$.

A multi-ciphertext scheme is (t, q, ε)-secure if $\mathbf{Adv}^{\mathrm{priv}}_{\Gamma}(A) \leq \varepsilon$, for any probabilistic adversary A that runs in time t and asks at most q queries.

4 Multiple Ciphertext Mode (MCM)

We propose a Multiple Ciphertext Mode (MCM) as a concrete instantiation of multi-ciphertext scheme. First, we describe so called basic MCM, with simple structure suitable for explanation and analysis of its properties. Then we discuss a more practical modification of the basic MCM. All constructions are based on counter mode (CTR) of block cipher and use ideas of secret sharing schemes.

4.1 Basic MCM

The key generation algorithm is trivial, we simply choose a random k-bit key K (it will be used for underlying block cipher). For basic MCM encryption we set $l = nT$. The tag is generated randomly and divided into n-bit vectors $\tau_1, \ldots, \tau_T$. These vectors are used as starting points for T independent streams of CTR blocks. Finally, we combine the CTR streams and message blocks to produce N ciphertexts. Let $M = (P_1, \ldots, P_m)$ be the input message consisting of m blocks. We denote by MCM(E) the basic MCM with an underlying transformation E.

Let us define N vectors $\alpha_1, \ldots, \alpha_N \in \mathrm{GF}(2^n)^{T+1}$ in the following way: $\alpha_i = (\mathrm{bin}(1), \mathrm{bin}(i), \mathrm{bin}(i)^2, \ldots, \mathrm{bin}(i)^T)$, where all computations are performed in $\mathrm{GF}(2^n)$.

Function $\mathcal{E}_K(P_1, \ldots, P_m)$:
$\tau = (\tau_1, \ldots, \tau_T) \xleftarrow{\$} (\{0,1\}^n)^T$
for $i = 1, \ldots, m$:
$(R_{i,1}, \ldots, R_{i,T}) \leftarrow (E_K(\tau_1 \oplus \mathrm{bin}(i)), \ldots, E_K(\tau_T \oplus \mathrm{bin}(i)))$
$\beta = (P_i, R_{i,1}, \ldots, R_{i,T})$
$(C_{1,i}, \ldots, C_{N,i}) \leftarrow (\langle \alpha_1, \beta \rangle, \ldots, \langle \alpha_N, \beta \rangle)$
return $(\underbrace{(C_{1,1}, \ldots, C_{1,m})}_{C_1}, \ldots, \underbrace{(C_{N,1}, \ldots, C_{N,m})}_{C_N}, \tau)$

Let us denote by α'_i the vector obtained from α_i by deleting the first element, i.e. $\alpha'_i = (\mathrm{bin}(i), \mathrm{bin}(i)^2, \ldots, \mathrm{bin}(i)^T)$. Decryption algorithm is straightforward:

Function $\mathcal{D}_K(C_j, j, \tau)$:
$(\tau_1, \ldots, \tau_T) \leftarrow \tau$
for $i = 1, \ldots, m$:
$(R_{i,1}, \ldots, R_{i,T}) \leftarrow (E_K(\tau_1 \oplus \mathrm{bin}(i)), \ldots, E_K(\tau_T \oplus \mathrm{bin}(i)))$
$\beta' = (R_{i,1}, \ldots, R_{i,T})$
$P_i = C_{j,i} \oplus \langle \alpha'_j, \beta' \rangle$
return $\underbrace{(P_1, \ldots, P_m)}_{M}$

Knowing at least $T+1$ ciphertexts (say $C_{j_1}, C_{j_2}, \ldots, C_{j_{T+1}}$) allows recovery of the original message M without knowing the key. We show that each plaintext block P_i (for $i = 1, \ldots, m$) can be recovered from corresponding blocks of ciphertexts: $C_{j_1,i}, \ldots, C_{j_{T+1},i}$. From encryption equations we have (as always, the computations are in $\mathrm{GF}(2^n)$):

$$
\begin{aligned}
C_{j_1,i} &= P_i \oplus \mathrm{bin}(j_1) R_{i,1} \oplus \mathrm{bin}(j_1)^2 R_{i,2} \oplus \ldots \oplus \mathrm{bin}(j_1)^T R_{i,T} \\
C_{j_2,i} &= P_i \oplus \mathrm{bin}(j_2) R_{i,1} \oplus \mathrm{bin}(j_2)^2 R_{i,2} \oplus \ldots \oplus \mathrm{bin}(j_2)^T R_{i,T} \\
&\ldots \\
C_{j_{T+1},i} &= P_i \oplus \mathrm{bin}(j_{T+1}) R_{i,1} \oplus \mathrm{bin}(j_{T+1})^2 R_{i,2} \oplus \ldots \oplus \mathrm{bin}(j_{T+1})^T R_{i,T}
\end{aligned}
$$

Thus, we have a system of $T+1$ linear equations with $T+1$ variables $P_i, R_{i,1} \ldots, R_{i,T}$. The matrix of the system (let us denote it $\mathbf{A}_J$ where $J = \{j_1, \ldots, j_{T+1}\}$) is Vandermonde matrix with distinct $\mathrm{bin}(j_1), \ldots, \mathrm{bin}(j_{T+1})$, and therefore the system has a unique solution. The algorithm Recover solves the system for each message block and computes the values of $P_1, \ldots, P_m$. The efficiency of Recover algorithm depends on how fast we can find an unknown P_i for particular system. The system of linear equations with Vandermonde matrix can be solved in time $O(T^2)$ [2], so the overall complexity of recovery is $O(mT^2)$.

However, we can do even better. We compute the inverse matrix A_J^{-1}. Then

$$
\underbrace{\mathbf{A}_J^{-1} \cdot \mathbf{A}_J}_{\mathbf{I}_{T+1}} \cdot \begin{pmatrix} P_i \\ R_{i,1} \\ \vdots \\ R_{i,T} \end{pmatrix} = \mathbf{A}_J^{-1} \cdot \begin{pmatrix} C_{j_1,i} \\ C_{j_2,i} \\ \vdots \\ C_{j_{T+1},i} \end{pmatrix}
$$

and P_i can be easily computed in $O(T)$ time: $P_i = \langle \gamma, (C_{j_1,i}, \ldots, C_{j_{T+1},i}) \rangle$, where γ is the first row of A_J^{-1}. We can compute $\mathbf{A}_J^{-1}$ in advance or at the beginning of the algorithm Recover (it does not depend on ciphertexts) in $O(T^2)$ time [6]. Therefore the overall complexity of recovery is $O(T^2 + mT)$. In practice, we can usually expect $m \gg T$, so the term T^2 is not an issue.

Remark 2. The lower bound for time complexity of Recover in general is $\Omega(mT)$ (use every block of every ciphertext). Otherwise the security properties of the scheme are violated.

4.2 Security of basic MCM

We start with analyzing the security of basic MCM in an ideal world, where the underlying block cipher $E_K(\cdot)$ is modeled as a random function $f \xleftarrow{\$} \mathrm{Func}(n, n)$. Certainly, a random permutation (instead of function) would be more precise for ideal model of $E_K(\cdot)$. However, the difference between random permutation and random function on $\{0,1\}^n$ is negligible (a hypothetical distinguisher has to find a collision). Therefore, for our ideal world analysis it suffices to take random f.

Theorem 1. *Let* $\mathrm{MCM}(f)$ *be a basic MCM where the underlying block cipher* E_K *is instantiated as* $f \xleftarrow{\$} \mathrm{Func}(n, n)$*. Let* A *be an adversary which asks at most* q *queries. Then* $\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{MCM}(f)}(A) \leq (q^2T^2m)/2^n$*.*

Proof. Let $(M^{(1)}, I^{(1)}), \ldots, (M^{(q)}, I^{(q)})$ be the queries that A asks its oracle. We will add a superscript "(s)" to variables in MCM to denote their value for particular query $(M^{(s)}, I^{(s)})$. Recall, A gets a subset $I^{(s)}$ of ciphertexts corresponding to the encryption of $M^{(s)}$ in the first scenario, and the same subset of ciphertexts corresponding to a randomly chosen plaintext in the second scenario.

We define an event Over as follows: Over occurs if there exists an arbitrary overlap of any CTR streams generated by MCM when computing the answers to A's queries. More precisely, Over happens if and only if there exist $s, s' \in \{1, \ldots, q\}$, $j, j' \in \{1, \ldots, T\}$, $i, i' \in \{1, \ldots, m\}$ such that $(s, j, i) \neq (s', j', i')$ and $\tau_j^{(s)} \oplus \mathrm{bin}(i) = \tau_{j'}^{(s')} \oplus \mathrm{bin}(i')$. Since $\tau^{(s)}$ values are generated randomly, the probability of Over is the same in both scenarios. Let us estimate the upper bound of the probability $\Pr[\mathsf{Over}]$. We can think of all $q \cdot T$ CTR streams as independent streams ($\tau^{(s)}$ are random). The probability of overlap when generating i-th stream is upper bounded (when previous streams do not overlap) as follows:

$$p_i \leq \frac{(i-1)(m-1) + (i-1)m}{2^n} \leq \frac{2m(i-1)}{2^n}$$

Then the upper bound of $\Pr[\mathsf{Over}] \leq \sum_{i=1}^{qT} p_i$ can be computed:

$$\Pr[\mathsf{Over}] \leq \sum_{i=1}^{qT} \frac{2m(i-1)}{2^n} \leq \frac{q^2T^2m}{2^n} \tag{1}$$

Let Win_1 (Win_2) be the event that A outputs 1 in the first (second) scenario. We need to estimate (upper bound) $\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{MCM}(f)}(A)$:

$$\begin{aligned}\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{MCM}(f)}(A) &= \Pr[\mathsf{Win}_1] - \Pr[\mathsf{Win}_2] \\ &= (\Pr[\mathsf{Win}_1 \mid \mathsf{Over}] \cdot \Pr[\mathsf{Over}] + \Pr[\mathsf{Win}_1 \mid \neg\mathsf{Over}] \cdot \Pr[\neg\mathsf{Over}]) \\ &\quad - (\Pr[\mathsf{Win}_2 \mid \mathsf{Over}] \cdot \Pr[\mathsf{Over}] + \Pr[\mathsf{Win}_2 \mid \neg\mathsf{Over}] \cdot \Pr[\neg\mathsf{Over}])\end{aligned}$$

If there is no overlap in the CTR streams, for each ciphertext block $C^{(s)}_{j,i}$ (for $j \in I^{(s)}$, $i \in \{0, \ldots, m\}$) which A receives as an oracle's answer the following holds: each value $R^{(s)}_{i,\cdot}$ used for encryption is random and independent of other values. Since A gets T ciphertexts, this yields the system of T linear equations with $T+1$ unknowns $P^{(s)}_i, R^{(s)}_{i,1}, \ldots, R^{(s)}_{i,T}$ ($T \times (T+1)$ Vandermonde coefficient matrix) – thus the corresponding plaintext block $P^{(s)}_i$ can be arbitrary. Therefore, A cannot distinguish the scenarios when there is no overlap, i.e.

$$\Pr[\mathsf{Win}_1 \mid \neg\mathsf{Over}] = \Pr[\mathsf{Win}_2 \mid \neg\mathsf{Over}] \ .$$

Let us continue with estimation of $\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{MCM}(f)}(A)$:

$$\begin{aligned}\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{MCM}(f)}(A) &= \Pr[\mathsf{Win}_1 \mid \mathsf{Over}] \cdot \Pr[\mathsf{Over}] - \Pr[\mathsf{Win}_2 \mid \mathsf{Over}] \cdot \Pr[\mathsf{Over}] \\ &= (\Pr[\mathsf{Win}_1 \mid \mathsf{Over}] - \Pr[\mathsf{Win}_2 \mid \mathsf{Over}]) \cdot \Pr[\mathsf{Over}] \\ &\leq \Pr[\mathsf{Over}]\end{aligned}$$

The proof is finished by combining this bound with (1). □

Remark 3. We use fixed-length messages (having m blocks) in the proofs of Theorem 1 and Theorem 2. This simplifies the presentation of the proofs. However, the proofs can be easily generalized for messages with variable number of blocks.

We move from the ideal world to more concrete security. We prove that the MCM(E) scheme is secure, assuming the block cipher is secure pseudo-random function.

Theorem 2. *Let F be a (t', q', ε')-secure pseudo-random function. Then the* MCM(F) *is (t, q, ε)-secure multi-ciphertext scheme, where $t = t' - cq'(1 + N/T)$, $q = q'/(mT)$ and $\varepsilon = 2\varepsilon' + q^2T^2m/2^n$ for some constant $c > 0$.*

Proof. Let A be an adversary that breaks MCM(F) scheme, i.e. $\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{MCM}(F)}(A) > \varepsilon$. We build a distinguisher D that attacks the pseudo-randomness of F:

Distinguisher $D^{O(\cdot)}$:
$b \xleftarrow{\$} \{0, 1\}$
run A and answer its oracle queries (M, I):
  simulate the encryption of M using $O(\cdot)$ to simulate the E_K calls
  return T ciphertexts according I
if $(b = b')$ **return** 1 **else return** 0

Distinguisher D has access to oracle $O(\cdot)$, instantiated as $F_K(\cdot)$ (for random key K) or a random function $f \xleftarrow{\$} \mathrm{Func}(n,n)$. If A makes q queries then D asks $q' = qmT$ queries. Moreover, if A runs in time t then D runs in time $t' = t + cqm(T+N) = t + cq'(1+N/T)$ for some constant $c > 0$, were the additional term is due to encryptions that D performs.

Let us estimate $\mathbf{Adv}^{\mathrm{prf}}_F(D)$. To shorten the length of probabilistic expressions we omit the random selections of K or f (they are known from the context). We have

$$\mathbf{Adv}^{\mathrm{prf}}_F(D) = \Pr[D^{F_K(\cdot)} \Rightarrow 1] - \Pr[D^{f(\cdot)} \Rightarrow 1]$$

We evaluate the first probability. Let us define the following shortcuts: $A(F_K) = A^{\mathcal{E}_K^{F_K}(\cdot)\downarrow(\cdot)}$, $A(F_K\$) = A^{\mathcal{E}_K^{F_K}(\$^{|\cdot|})\downarrow(\cdot)}$). Then

$$\begin{aligned}
\Pr[D^{F_K(\cdot)} \Rightarrow 1] &= \Pr[A(F_K) \Rightarrow 1 \wedge b = 1] - \Pr[A(F_K\$) \Rightarrow 0 \wedge b = 0] \\
&= \frac{1}{2} \cdot \Pr[A(F_K) \Rightarrow 1 \mid b = 1] - \frac{1}{2} \cdot \Pr[A(F_K\$) \Rightarrow 0 \mid b = 0] \\
&= \frac{1}{2} \cdot (\Pr[A(F_K) \Rightarrow 1] - \Pr[A(F_K\$) \Rightarrow 0]) \\
&= \frac{1}{2} + \frac{1}{2} \cdot \mathbf{Adv}^{\mathrm{priv}}_{\mathrm{MCM}(F)}(A)
\end{aligned}$$

Similarly

$$\Pr[D^{f(\cdot)} \Rightarrow 1] = \frac{1}{2} + \frac{1}{2} \cdot \mathbf{Adv}^{\mathrm{priv}}_{\mathrm{MCM}(f)}(A) \ .$$

Putting this together we have

$$\mathbf{Adv}^{\mathrm{prf}}_F(D) = \frac{1}{2} \cdot \mathbf{Adv}^{\mathrm{priv}}_{\mathrm{MCM}(F)}(A) - \frac{1}{2} \cdot \mathbf{Adv}^{\mathrm{priv}}_{\mathrm{MCM}(f)}(A) \ .$$

Using the fact that $\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{MCM}(F)}(A) > \varepsilon$ and Theorem 1 we get

$$\mathbf{Adv}^{\mathrm{prf}}_F(D) > \frac{\varepsilon}{2} - \frac{q^2T^2m}{2^{n+1}} = \varepsilon',$$

a contradiction with assumption that F is (t', q', ε')-secure pseudo-random function. □

5 MCM Variants

There are several directions how the basic MCM can be improved. We discuss some possibilities in this section.

5.1 Efficiency of Encryption and Recovery

The encryption algorithm requires computation of T CTR streams that are combined with the message blocks to obtain N ciphertexts. Notice that knowing

the key and tag τ allows a distributed computation of particular ciphertexts. This can be useful for an application, where a distant backup location computes just "its own" ciphertext $C_i = \mathcal{E}_K^i(M)$ (and forgets the key afterwards, for example).

The need to compute T CTR streams makes the encryption approximately T times slower than simple symmetric encryption with the same block cipher. Here we assume that the linear combinations of CTR streams can be computed much faster than performing the block cipher transformation (otherwise the slowdown in even worse). In situations where a more efficient computation of particular ciphertext is required, we can choose a different set linear combinations. More importantly, different set of linear combinations can reduce the performance of Recover algorithm.

To be more precise, let $\mathbf{A}$ be an $N \times (T+1)$ matrix over $\mathrm{GF}(2^n)$ that defines the linear combinations for producing ciphertexts. For the basic MCM scheme we have ($i \in \{1, \ldots, m\}$):

$$\begin{pmatrix} C_{1,i} \\ \vdots \\ C_{N,i} \end{pmatrix} = \mathbf{A} \cdot \begin{pmatrix} P_i \\ R_{i,1} \\ \vdots \\ R_{i,T} \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{pmatrix} \cdot \begin{pmatrix} P_i \\ R_{i,1} \\ \vdots \\ R_{i,T} \end{pmatrix} \quad .$$

However, we can choose a different matrix as long as it guarantees the properties of the scheme. An zero element in j-th row of a matrix means that particular CTR stream is not needed to compute C_j.

Example 1. Let us illustrate this with $T = 2$ and $N = 3$ (one can expect that N will not be very large in practical scenarios). Then, for example, the following matrix $\mathbf{A}$ defines an alternative computation of ciphertexts:

$$\mathbf{A} = \begin{pmatrix} 1\ 1\ 0 \\ 1\ 0\ 1 \\ 1\ 1\ 1 \end{pmatrix} \qquad \Rightarrow \qquad \begin{cases} C_{1,i} = P_i \oplus R_{i,1} \\ C_{2,i} = P_i \oplus R_{i,2} \\ C_{3,i} = P_i \oplus R_{i,1} \oplus R_{i,2} \end{cases}$$

It is easy to verify that knowing any pair of ciphertexts does not help in computing the plaintext. On the other hand, three ciphertexts allow very fast recovery of plaintext without knowing the key: $P_i = C_{1,i} \oplus C_{2,i} \oplus C_{i,3}$. Moreover, the ciphertext C_1 and C_2 can be computed as fast as standard encryption in CTR mode.

5.2 Shorter Tags

The basic MCM scheme generates the tag τ randomly as nT-bit string. We can shorten the tag to n bits by employing a pseudo-random function. Let $F : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a pseudo-random function (in practice we can use the underlying block cipher E). We modify the basic MCM scheme as follows:

– The key generation algorithm chooses two independent random k-bit keys K, K'. The key K is used in computation of CTR stream as before, and K' will be used for expansion of shortened tag τ.

– The encryption algorithm starts with the following code to produce τ and $\tau_1, \ldots, \tau_T$:

$$\tau \xleftarrow{\$} \{0,1\}^n$$
$$(\tau_1, \ldots, \tau_T) \leftarrow (F_{K'}(\tau \oplus \mathrm{bin}(1)), \ldots, F_{K'}(\tau \oplus \mathrm{bin}(T)))$$

  The rest of the algorithm (computation of ciphertexts) remains intact.
– The decryption algorithm must be adjusted accordingly, i.e. $\tau_1, \ldots, \tau_T$ are computed from τ as shown above.

This modification has no impact on Recover algorithm.

It can be easily verified that Theorem 1 holds for this modification, and Theorem 2 holds with insignificant adjustments: $t' = t - cq'\left(1 + \frac{mN}{(m+1)T}\right)$ and $q' = \frac{q}{(m+1)T}$.

5.3 Authentication

In many practical situation some form of authenticity is often required in addition to confidentiality provided by encryption. The multiple ciphertext mode can be extended to provide authentication. We can use a two-pass construction similar to those used for computation of authentication tag in double ciphertext mode [3] or synthetic initialization vector mode [5].

The construction uses a pseudo-random functions $G : \{0,1\}^k \times \{0,1\}^{nm} \to \{0,1\}^n$ and $F : \{0,1\}^k \times \{0,1\}^{nm} \to \{0,1\}^n$. We extend the key generation algorithm to produce three k-bit random keys K_G, K_F and K. The first key K_G is used for computing authentication tag τ_G used in place of the original tag τ. The second key K_F is used for producing tags for CTR streams (taking a construction from Sect. 5.2). The key K is used as in basic MCM. Therefore the encryption algorithm is modified as follows:

Function $\mathcal{E}_{K_G,K_F,K}(P_1, \ldots, P_m)$:
$\tau_G = G_{K_G}(P_1 \ldots P_m)$
$(\tau_1, \ldots, \tau_T) \leftarrow (F_{K_F}(\tau_G \oplus \mathrm{bin}(1)), \ldots, F_{K_F}(\tau_G \oplus \mathrm{bin}(T)))$
for $i = 1, \ldots, m$:
  $(R_{i,1}, \ldots, R_{i,T}) \leftarrow (E_K(\tau_1 \oplus \mathrm{bin}(i)), \ldots, E_K(\tau_T \oplus \mathrm{bin}(i)))$
  $\beta = (P_i, R_{i,1}, \ldots, R_{i,T})$
  $(C_{1,i}, \ldots, C_{N,i}) \leftarrow (\langle \alpha_1, \beta \rangle, \ldots, \langle \alpha_N, \beta \rangle)$
return $(C_1, \ldots, C_N, \tau_G)$

The decryption algorithm checks for validity of authentication tag τ_G (beside decrypting the message). In case the authenticity cannot be verified the decryption returns $\perp$. Analogous verification must be added into Recover algorithm.

Function $\mathcal{D}_{K_G,K_F,K}(C_j, j, \tau_G)$:
$(\tau_1, \ldots, \tau_T) \leftarrow (F_{K_F}(\tau_G \oplus \mathrm{bin}(1)), \ldots, F_{K_F}(\tau_G \oplus \mathrm{bin}(T)))$
for $i = 1, \ldots, m$:
$(R_{i,1}, \ldots, R_{i,T}) \leftarrow (E_K(\tau_1 \oplus \mathrm{bin}(i)), \ldots, E_K(\tau_T \oplus \mathrm{bin}(i)))$
$\beta' = (R_{i,1}, \ldots, R_{i,T})$
$P_i = C_{j,i} \oplus \langle \alpha'_j, \beta' \rangle$
$M \leftarrow (P_1, \ldots, P_m)$
if $(G_{k_G}(M) = \tau_G)$ **return** M
return $\perp$

6 Conclusion

We presented and proved the properties of basic multiple ciphertext mode. We discussed some extensions of the basic construction. Interesting open problems for further research are: generalization of variant described in Sect. 5.1; and design of one-pass mode that si (T, N) multi-ciphertext scheme and simultaneosly guarantees the authenticity of the ciphertexts.

Acknowledgement. This work was supported by VEGA 1/0266/09.

References

1. M. Bellare, A. Desai, E. Jokipi, P. Rogaway: *A Concrete Security Treatment of Symmetric Encryption*, Proceedings of the 38th Symposium on Foundations of Computer Science, IEEE, pp. 394–403, 1997.
2. Å. Björck, V. Pereyra: *Solution of Vandermonde Systems of Equations*, Mathematics of Computation, Vol. 24, No. 112, pp. 893–903, American Mathematical Society, 1970.
3. D. Chakraborty, C. Mancillas-López: *Double Ciphertext Mode: A Proposal for Secure Backup*, Cryptology ePrint Archive, Report No. 2010/369, 2010.
4. J. Katz, Y. Lindell: *Introduction to Modern Cryptography*, Chapman & Hall/CRC, 2008.
5. P. Rogaway, T. Shrimpton: *A Provable-Security Treatment of the Key-Wrap Problem*, Advances in Cryptology – EUROCRYPT 2006, Lecture Notes in Computer Science, vol. 4004, Springer, pp. 373–390, 2006.
6. J. Traub: *Associated Polynomials and Uniform Methods for the Solution of Linear Problems*, SIAM Review, Vol. 8, No. 3, pp. 277–301, Society for Industrial and Applied Mathematics, 1966.