POLITECHNIKA GDAŃSKA
WYDZIAŁ ELEKTRONIKI, TELEKOMUNIKACJI I INFORMATYKI

Imię i nazwisko autora rozprawy: **Bartosz Czaplewski**
Dyscyplina naukowa: Telekomunikacja

ROZPRAWA DOKTORSKA

Tytuł rozprawy w języku polskim: Nowe metody łącznego fingerprintingu i deszyfracji do zabezpieczania obrazów kolorowych

Tytuł rozprawy w języku angielskim: New joint fingerprinting and decryption methods for securing color images

Promotor *podpis*	Drugi promotor *podpis*
dr hab. inż. Roman Rykaczewski	
Promotor pomocniczy *podpis*	Kopromotor *podpis*

Gdańsk, rok 2015

SPIS TREŚCI

WYKAZ OZNACZEŃ 5
WYKAZ SKRÓTÓW 11
1. WSTĘP 13
1.1. Cel i teza pracy 16
1.2. Struktura rozprawy 17
2. OCHRONA MULTIMEDIÓW POPRZEZ FINGERPRINTING 19
2.1. Klasyczne metody działania fingerprintingu 19
2.2. Kryteria konstrukcji zbioru fingerprintów 22
2.3. Wymagania dla metod fingerprintingu 24
2.4. Umiejscowienie fingerprintingu i wykorzystanie transmisji multicast 26
2.5. Podsumowanie 28
3. ZAGROŻENIA DLA OSADZONYCH FINGERPRINTÓW 29
3.1. Ataki realizowane przez pojedynczego pirata 30
3.2. Ataki zmowy 31
3.3. Identyfikacja piratów w ataku zmowy 34
3.4. Osadzanie fingerprintów w dziedzinie częstotliwości 36
3.5. Podsumowanie 38
4. ŁĄCZNY FINGERPRINTING I DESZYFRACJA 39
4.1. Zasady działania metod JFD 39
4.2. Bezpieczeństwo szyfrowania w JFD 42
4.3. Przegląd literatury metod JFD 44
4.4. Podsumowanie 48
5. METODA JFD OPARTA NA ILOCZYNIE MACIERZY 49
5.1. Konstrukcja kluczy 49
5.2. Szyfrowanie 50
5.3. Deszyfrowanie 53
5.4. Identyfikacja piratów 54
5.5. Okresowa zmiana klucza 56
5.6. Podsumowanie 58
6. PODSTAWY RACHUNKU KWATERNIONOWEGO 59
6.1. Algebra kwaternionów 59
6.2. Przekształcenia punktu w przestrzeni trójwymiarowej 62
6.3. Podsumowanie 63
7. METODA JFD OPARTA NA ROTACJI KWATERNIONÓW Z CQC 65
7.1. Konstrukcja kluczy 66
7.2. Szyfrowanie 67
7.3. Deszyfrowanie 70
7.4. Identyfikacja piratów 71
7.5. Podsumowanie 73
8. METODA JFD OPARTA NA SERII ROTACJI KWATERNIONÓW 75
8.1. Konstrukcja kluczy 75
8.2. Szyfrowanie 77

8.3. Deszyfrowanie ... 81
8.4. Identyfikacja piratów ... 83
8.5. Podsumowanie ... 85
9. WYNIKI BADAŃ ... 87
9.1. Parametry symulacji ... 87
9.2. Niezauważalność osadzonych fingerprintów ... 88
9.3. Odporność na szum ... 101
9.4. Odporność na kompresję ... 103
9.5. Odporność na atak zmowy ... 105
9.6. Odporność na atak zmowy połączony z kompresją ... 112
9.7. Porównanie i ocena nowych metod JFD ... 116
9.8. Podsumowanie ... 120
10. SPRAWIEDLIWY PROTOKÓŁ DYSTRYBUCJI DLA METOD JFD ... 123
10.1. Problem braku ochrony klienta ... 123
10.2. Proponowane rozwiązanie ... 124
10.3. Podsumowanie ... 132
11. PODSUMOWANIE ... 133
BIBLIOGRAFIA ... 137
WYKAZ RYSUNKÓW I TABEL ... 143
ZAŁĄCZNIKI ... 147

WYKAZ OZNACZEŃ

α – współczynnik siły osadzania fingerprintów
β – parametr szyfrowania w metodzie JFD opartej o serię rotacji kwaternionów
γ – kąt rotacji
ε – parametr ograniczający prawdop. błędu dla kodów Tardos i Boneh-Shaw
$\rho^{(u)}$ – współczynnik korelacji między fingerprintem pirata, a fingerprintem u-tego użytk.
ψ_1 – funkcja generująca składową fingerprintu klienta związaną z zamówieniem
ψ_2 – funkcja generująca fingerprint klienta
ψ_3 – funkcja generująca klucz deszyfrujący klienta
a – parametr pomocniczy
a_x – pierwsza współrzędna wektora opisującego płaszczyznę rotacji
a_y – druga współrzędna wektora opisującego płaszczyznę rotacji
a_z – trzecia współrzędna wektora opisującego płaszczyznę rotacji
A – liczba niesłusznie oskarżonych niewinnych użytkowników
arg – ogólny argument, dot. opisu protokołu
ack – potwierdzenie przyjęcia zamówienia usługi, dot. opisu protokołu
b – indeks rozpatrywanego bloku DCT obrazu
B – liczba bloków DCT obrazu
c – indeks rozpatrywanego pirata w zmowie
C – liczba piratów w zmowie
$\mathrm{Cert_{CA}}$ – certyfikat urzędu certyfikacji, dot. opisu protokołu
$\mathrm{Cert_{K}}$ – certyfikat klienta, dot. opisu protokołu
$\mathrm{Cert_{KGC}}$ – certyfikat centrum generowania kluczy, dot. opisu protokołu
$\mathrm{Cert_{PIC}}$ – certyfikat centrum identyfikacji piratów, dot. opisu protokołu
$\mathrm{Cert_{S}}$ – certyfikat sprzedawcy, dot. opisu protokołu
$d_l^{(u)}$ – l-ty kwaternion wektora klucza deszyfrującego u-tego użytkownika
$d_{r_1,r_2}^{(u)}$ – kwaternion w wierszu r_1 i kolumnie r_2 klucza deszyfrującego u-tego użytkownika
$\mathbf{d}^{(u)}$ – wektor klucza deszyfrującego u-tego użytkownika
$\mathbf{D}^{(u)}$ – macierz klucza deszyfrującego u-tego użytkownika
D – klucz deszyfrujący metody JFD, dot. opisu protokołu
e_l – l-ty kwaternion wektora klucza szyfrującego
e_{r_1,r_2} – kwaternion w wierszu r_1 i kolumnie r_2 klucza klucza szyfrującego
$\mathbf{e}$ – wektor klucza szyfrującego
$\mathbf{E}$ – macierz klucza szyfrującego
E – klucz szyfrujący metody JFD, dot. opisu protokołu
$\mathrm{Encr_{p\text{-}K}}$ – szyfrowanie kluczem publicznym klienta, dot. opisu protokołu
$\mathrm{Encr_{p\text{-}PIC}}$ – szyfrowanie kluczem publ. centrum identyfikacji piratów, dot. opisu protokołu

$f^{(u)}$ – fingerprint *u*-tego użytkownika

$f^{(u)\prime}$ – wyskalowany fingerprint *u*-tego użytkownika

$f_{u,m}$ – element z *u*-tego wiersza i *m*-tej kolumny macierzy zbioru fingerprintów $\mathbf{F}$

f_m^{P} – *m*-ty element fingerprintu wydobytego z pirackiej kopii

$f_r^{(u)}$ – *r*-ty kwaternion wektora fingerprintu *u*-tego użytkownika

$f_{r_1,r_2}^{(u)}$ – kwaternion w wierszu r_1 i kolumnie r_2 macierzy fingerprintu *u*-tego użytkownika

$\mathbf{f}^{(u)}$ – wektor kwaternionów fingerprintu *u*-tego użytkownika

$\mathbf{f}_{\mathrm{P}}$ – wektor kwaternionów fingerprintu wydobytego z pirackiej kopii

$\overline{\mathbf{f}_{\mathrm{P}}}$ – wartość średnia współczynników wszystkich składowych urojonych w $\mathbf{f}_{\mathrm{P}}$

$\mathbf{f}_{\mathrm{emb}}^{(u)}$ – wektor kwaternionów fingerprintu wydobytego z obrazu *u*-tego użytkownika

$\overline{\mathbf{f}_{\mathrm{emb}}^{(u)}}$ – wartość średnia współczynników wszystkich składowych urojonych w $\mathbf{f}_{\mathrm{emb}}^{(u)}$

$\mathbf{F}^{(u)}$ – macierz fingerprintu *u*-tego użytkownika

$\mathbf{F}_{\mathrm{P}}$ – macierz fingerprintu wydobytego z pirackiej kopii

$\overline{\mathbf{F}_{\mathrm{P}}}$ – wartość średnia wszystkich składowych macierzy $\mathbf{F}_{\mathrm{P}}$

$\mathbf{F}_{\mathrm{emb}}^{(\mathrm{u})}$ – macierz właściwego osadzonego fingerprintu w kopii *u*-tego użytkownika

$\overline{\mathbf{F}_{\mathrm{emb}}^{(\mathrm{u})}}$ – wartość średnia wszystkich składowych macierzy $\mathbf{F}_{\mathrm{emb}}^{(\mathrm{u})}$

F – fingerprint klienta w opisie protokołu

F_{K} – składowa fingerprintu klienta pochodząca od klienta, dot. opisu protokołu

F_{ord} – składowa fingerprintu klienta związana z zamówieniem, dot. opisu protokołu

F_{S} – składowa fingerprintu klienta pochodząca od sprzedawcy, dot. opisu protokołu

g – pomocnicza zmienna indeksowania

$g_1(z)$ – pomocnicza zmienna indeksowania, zależna od indeksu rozpatrywanej rotacji *z*

$g_2(z)$ – pomocnicza zmienna indeksowania, zależna od indeksu rozpatrywanej rotacji *z*

h – pomocnicza zmienna indeksowania

h_1 – pomocnicza zmienna indeksowania

h_2 – pomocnicza zmienna indeksowania

i – pierwsza jednostka urojona

I – liczba poprawnie zidentyfikowanych piratów

$\mathbf{I}$ – macierz jednostkowa

j – druga jednostka urojona

k – trzecia jednostka urojona

K_{E} – klucz szyfrujący

K_{D} – klucz deszyfrujący

$K_{\mathrm{D}}^{(u)}$ – klucz deszyfrujący *u*-tego użytkownika

l – pomocnicza zmienna indeksowania

m – indeks elementu fingerprintu

M – długość fingerprintu, długość ciągu binarnego
n – indeks rozpatrywanego elementu obrazu
n_1 – indeks rozpatrywanego wiersza w macierzy obrazu
n_2 – indeks rozpatrywanej kolumny w macierzy obrazu
n_{A} – zakłócenie addytywne
n_{B} – zakłócenie addytywne przy detekcji niekoherentnej
N – liczba elementów obrazu wybranych do szyfrowania selektywnego
N_1 – liczba wierszy macierzy obrazu
N_2 – liczba kolumn macierzy obrazu
ord – zamówienie usługi, dot. opisu protokołu
p – przykładowy kwaternion
p_w – składowa rzeczywista kwaternionu p
p_x – pierwsza składowa urojona kwaternionu p
p_y – druga składowa urojona kwaternionu p
p_z – trzecia składowa urojona kwaternionu p
$\vec{p}_v$ – wektorowy człon kwaternionu p
p^* – sprzężenie kwaternionu p
p_{IN} – kwaternion reprezentujący punkt w przestrzeni 3D przed przekształceniem
p_{OUT} – kwaternion reprezentujący punkt w przestrzeni 3D po przekształceniu
P_f – średnia moc fingerprintu
P_x – średnia moc sygnału hosta
P_n – średnia moc zakłóceń
$\mathrm{P_R}$ – prawdopodobieństwo wykorzystywane w losowym ataku różnicowym
$\mathrm{P_{TP}}$ – prawdopodobieństwo poprawnie pozytywnej (ang. *true positive*) detekcji
$\mathrm{P_{FP}}$ – prawdopodobieństwo fałszywie pozytywnej (ang. *false positive*) detekcji
$\mathrm{P_{TP}^{(A)}}$ – prawdopodobieństwo poprawnie pozytywnej detekcji dla detektora A
$\mathrm{P_{FP}^{(A)}}$ – prawdopodobieństwo fałszywie pozytywnej detekcji dla detektora A
$\mathrm{P_{TP}^{(B)}}$ – prawdopodobieństwo poprawnie pozytywnej detekcji dla detektora B
$\mathrm{P_{FP}^{(B)}}$ – prawdopodobieństwo fałszywie pozytywnej detekcji dla detektora B
$\mathbf{P}_l$ – l-ta macierz permutacji
pir – tożsamość piratów, dot. opisu protokołu
prf – dowód posiadania prawa do dystrybucji multimediów, dot. opisu protokołu
q – przykładowy kwaternion
q_1 – przykładowy kwaternion
q_2 – przykładowy kwaternion
q_3 – przykładowy kwaternion

q_w – składowa rzeczywista kwaternionu q

q_x – pierwsza składowa urojona kwaternionu q

q_y – druga składowa urojona kwaternionu q

q_z – trzecia składowa urojona kwaternionu q

$\vec{q}_v$ – wektorowy człon kwaternionu q

q^* – sprzężenie kwaternionu q

$\|q\|$ – norma kwaternionu q

q^{-1} – odwrotność kwaternionu q

Q – jakość obrazu skompresowanego wyrażona w procentach

r – indeks rozpatrywanego elementu klucza

r_1 – indeks rozpatrywanego wiersza w macierzy klucza oraz macierzy fingerprintu

r_2 – indeks rozpatrywanej kolumny w macierzy klucza oraz macierzy fingerprintu

R – rozmiar klucza szyfrującego (rozmiar wektora lub macierzy kwadratowej)

R_1 – liczba wierszy macierzy klucza oraz macierzy fingerprintu

R_2 – liczba kolumn macierzy klucza oraz macierzy fingerprintu

res – wyniki identyfikacji piratów, dot. opisu protokołu

S – liczba zrealizowanych symulacji

$\mathrm{Sign}_{\mathrm{CA}}$ – podpis cyfrowy urzędu certyfikacji, dot. opisu protokołu

$\mathrm{Sign}_{\mathrm{K}}$ – podpis cyfrowy klienta, dot. opisu protokołu

$\mathrm{Sign}_{\mathrm{KGC}}$ – podpis cyfrowy centrum generowania kluczy, dot. opisu protokołu

$\mathrm{Sign}_{\mathrm{PIC}}$ – podpis cyfrowy centrum identyfikacji piratów, dot. opisu protokołu

$\mathrm{Sign}_{\mathrm{S}}$ – podpis cyfrowy sprzedawcy, dot. opisu protokołu

t_{D} – próg detekcji (identyfikacji) piratów

u – indeks rozpatrywanego użytkownika

U – liczba użytkowników

x – sygnał hosta, obraz oryginalny

x_{test} – sygnał testowy, piracka kopia

$x^{(u)}$ – oznakowana kopia danych *u*-tego użytkownika

$x_{\mathrm{test}}^{(u)}$ – sygnał testowy, zmodyfikowana oznakowana kopia danych *u*-tego użytkownika

$x_n^{(c)}$ – *n*-ta składowa oznakowanej kopii *c*-tego pirata

x_n^{ave} – *n*-ta składowa pirackiej kopii w ataku uśredniającym

x_n^{min} – *n*-ta składowa pirackiej kopii w ataku minimum

x_n^{max} – *n*-ta składowa pirackiej kopii w ataku maksimum

x_n^{med} – *n*-ta składowa pirackiej kopii w ataku medianowym

x_n^{minmax} – *n*-ta składowa pirackiej kopii w ataku min-max

x_n^{modneg} – *n*-ta składowa pirackiej kopii w zmodyfikowanym ataku różnicowym

x_n^{randneg} – *n*-ta składowa pirackiej kopii w losowym ataku różnicowym

x_{EF} – zbiór perceptualnie istotnych elementów obrazu

x_{NEF} – zbiór perceptualnie nieistotnych elementów obrazu

$x_{1,1}$ – element macierzy obrazu $\mathbf{X}$ wybrany do szyfrowania selektywnego

$x_{1,2}^{(b)}$ – współczynnik DCT z 1. wiersza i 2. kolumny *b*-tego bloku luminancji obrazu

$x_{1,1}^{(u)}$ – element macierzy obrazu odszyfrowanego $\mathbf{X}^{(u)}$ *u*-tego użytkownika

$x_{1,2}^{(u)(b)}$ – współczynnik z 1. wiersza i 2. kolumny b-tego bloku obrazu *u*-tego użytkownika

x_n – *n*-ty kwaternion obrazu

$x_n^{(u)}$ – *n*-ty kwaternion obrazu odszyfrowanego z fingerprintem *u*-tego użytkownika

$x_n^{(\text{red})}$ – *n*-ty współczynnik DCT czerwonej składowej obrazu

$x_n^{(\text{green})}$ – *n*-ty współczynnik DCT zielonej składowej obrazu

$x_n^{(\text{blue})}$ – *n*-ty współczynnik DCT niebieskiej składowej obrazu

$x_{n_1,n_2}^{(\text{red})}$ – współczynnik DCT czerwonej składowej obrazu

$x_{n_1,n_2}^{(\text{green})}$ – współczynnik DCT zielonej składowej obrazu

$x_{n_1,n_2}^{(\text{blue})}$ – współczynnik DCT niebieskiej składowej obrazu

$\mathbf{x}$ – wektor kwaternionów obrazu

$\mathbf{x}^{(\text{red})}$ – wektor wybranych współczynników DCT czerwonej składowej obrazu

$\mathbf{x}^{(\text{green})}$ – wektor wybranych współczynników DCT zielonej składowej obrazu

$\mathbf{x}^{(\text{blue})}$ – wektor wybranych współczynników DCT niebieskiej składowej obrazu

$\mathbf{x}^{(u)}$ – wektor kwaternionów obrazu odszyfrowanego z fingerprintem *u*-tego użytk.

$\mathbf{x}_{\text{P}}$ – wektor kwaternionów obrazu pirackiego

$\mathbf{X}$ – macierz obrazu

$\mathbf{X}^{(b)}$ – *b*-ty blok DCT luminancji obrazu

$\mathbf{X}^{(u)}$ – macierz obrazu odszyfrowanego z fingerprintem *u*-tego użytkownika

$\mathbf{X}^{(u)(b)}$ – *b*-ty blok DCT luminancji obrazu odszyfrowanego *u*-tego użytkownika

$\mathbf{X}^{(\text{red})}$ – macierz wybranych współczynników DCT czerwonej składowej obrazu

$\mathbf{X}^{(\text{green})}$ – macierz wybranych współczynników DCT zielonej składowej obrazu

$\mathbf{X}^{(\text{blue})}$ – macierz wybranych współczynników DCT niebieskiej składowej obrazu

$\mathbf{X}_{\text{P}}$ – macierz pirackiej kopii obrazu

X – obraz oryginalny, dot. opisu protokołu

y – dane zaszyfrowane

$y_{1,1}$ – element macierzy obrazu zaszyfrowanego $\mathbf{Y}$

$y_{1,2}^{(b)}$ – współczynnik z 1. wiersza i 2. kolumny *b*-tego bloku obrazu zaszyfrowanego

y_n – *n*-ty zaszyfrowany kwaternion obrazu

$y_n^{(\text{red})}$ – *n*-ty zaszyfrowany współczynnik DCT czerwonej składowej obrazu

$y_n^{(\text{green})}$ – *n*-ty zaszyfrowany współczynnik DCT zielonej składowej obrazu

$y_n^{(\text{blue})}$ – *n*-ty zaszyfrowany współczynnik DCT niebieskiej składowej obrazu

$y_{n_1,n_2}^{(\text{red})}$ – zaszyfrowany współczynnik DCT czerwonej składowej obrazu

$y_{n_1,n_2}^{(\text{green})}$ – zaszyfrowany współczynnik DCT zielonej składowej obrazu

$y_{n_1,n_2}^{(\text{blue})}$ – zaszyfrowany współczynnik DCT niebieskiej składowej obrazu

$\mathbf{y}$ – wektor zaszyfrowanych kwaternionów obrazu

$\mathbf{Y}$ – macierz obrazu zaszyfrowanego

$\mathbf{Y}^{(b)}$ – *b*-ty blok DCT luminancji obrazu zaszyfrowanego

Y – obraz zaszyfrowany zgodnie z wykorzystywaną metodą JFD, dot. opisu protokołu

z – indeks rozpatrywanej rotacji w serii rotacji kwaternionów

z_1 – kwaternion przesunięcia

z_2 – baza arytmetyki modularnej dla sumy kwaternionów

Z – długość serii rotacji kwaternionów

WYKAZ SKRÓTÓW

3D – przestrzeń trójwymiarowa (ang. Three-dimensional space)
AC – współczynnik „częstotliwościowy" w bloku DCT (ang. *Alternating Current*)
AES – zaawansowany standard szyfrowania (ang. *Advanced Encryption Standard*)
BIBD – metoda podziału zbioru (ang. *Balanced Incomplete Block Design*)
CA – urząd certyfikacji (ang. *Certificate Authority*)
CBC – tryb wiązania bloków szyfrogramów (ang. *Cipher-Block Chaining*)
CDMA – metoda kodowego wielodostępu do medium (ang. *Code Division Multiple Access*)
CQC – tryb wiązania zaszyfrowanych kwaternionów (ang. *Cipher-Quaternion Chaining*)
DCT – dyskretna transformacja kosinusowa (ang. *Discrete Cosine Transform*)
DC – współczynnik „stały" w bloku DCT (ang. *Direct Current*)
DES – standard szyfrowania danych (ang. *Data Encryption Standard*)
DRM – zarządzanie prawami cyfrowymi (ang. *Digital Rights Management*)
DWT – dyskretna transformacja falkowa (ang. *Discrete Wavelet Transform*)
EF – perceptualnie istotne elementy obrazu (ang. *Essential Features*)
FNR – stosunek poziomu fingerprintu do szumu (ang. *Fingerprint-to-Noise Ratio*)
FP – fałszywie pozytywna decyzja (ang. *False Positive*)
HVS – układ wzrokowy człowieka (ang. *Human Visual System*)
IDCT – odwrotna dyskretna transformacja kosinusowa (ang. *Inverse Discrete Cosine Transform*)
JFD – łączny fingerprinting i deszyfracja (ang. *Joint Fingerprinting and Decryption*)
JFE – łączny fingerprinting i szyfrowanie (ang. *Joint Fingerprinting and Encryption*)
JND – próg zauważalnej różnicy (ang. *Just Noticeable Difference*)
JPEG – metoda kompresji obrazów rastrowych (ang. *Joint Photographic Experts Group*)
JWE – łączny watermarking i szyfrowanie (ang. *Joint Watermarking and Encryption*)
KGC – centrum generowania kluczy (ang. *Key Generation Center*)
LFSR rejestr przesuwny z liniowym sprzężeniem zwrotnym (ang. *Linear Feedback Shift Register*)
LUT – tablica odniesienia (ang. *Lookup-Table*)
MA – założenie znakowania (ang. *Marking Assumption*)
MPEG – standard kodowania audio i wideo (ang. *Moving Picture Experts Group*)
MV – wektor ruchu (ang. *Motion Vector*)
NEF – perceptualnie nieistotne elementy obrazu (ang. *Non-Essential Features*)
PIC – centrum identyfikacji piratów (ang. *Pirate Identification Center*)
PKI – infrastruktura klucza publicznego (ang. *Public Key Infrastructure*)
PSNR – szczytowy stosunek sygnału do szumu (ang. *Peak Signal-to-Noise Ratio*)
RGB – model przestrzeni barw wykorzystujący 3 kanały kolorów (ang. *Red-Green-Blue*)
RLE – kodowanie długości serii (ang. *Run-Length Encoding*)
ROC – charakterystyka operacyjna detektora (ang. *Receiver Operating Characteristics*)
SNR – stosunek sygnału do szumu (ang. *Signal-to-Noise Ratio*)
STB – urządzenie dekodujące (ang. *Set-Top Box*)
SVD – dekompozycja macierzy według wartości osobliwych (ang. *Singular Value De-*

composition)

TP – bezbłędna decyzja pozytywna (ang. *True Positive*)

WHIM – multicast ze znakiem wodnym za pomocą hierarchii pośredników (ang. *Watermarking Multicast with a Hierarchy of Intermediaries*)

VoD – wideo na żądanie (ang. *Video on Demand*)

VQ – kwantyzacja wektorowa (ang. *Vector Quantization*)

YCbCr – model przestrzeni barw używający 1. kanału luminancji i 2. kanałów chrominancji

1. WSTĘP

We współczesnym świecie, dane multimedialne stały się integralną częścią naszego codziennego życia. Multimedia nie są jedynie istotnym narzędziem komunikacji, które umożliwia intuicyjną wymianę ważnych informacji, ale są także potrzebnym narzędziem ekspresji, które dzięki kreatywności autorów sprzyja rozwojowi kulturowemu. Informacja oraz sztuka, dla których multimedia stanowią medium, mają wartość ekonomiczną, czasem osobistą, i zazwyczaj mają silny wpływ na ogólny dobrobyt społeczeństwa. Możliwy charakter i przeznaczenie treści multimedialnych zawiera się w bardzo szerokim spektrum, które sięga od rozrywki, poprzez edukację i treści informacyjne, aż do zastosowań biznesowych.

Przy nieustannie rozwijanym i coraz powszechniejszym szerokopasmowym dostępie do Internetu, naturalnym stało się, że treści multimedialne codziennie są przesyłane przez sieć. Obecnie, dystrybucja multimediów w sieci przyjmuje rozmaite formy, np. usług interaktywnych, takich jak wideorozmowa i wideokonferencja, lub usług rozsiewczych, jak wideo na żądanie VoD (ang. *Video on Demand*) i wideo strumieniowe (ang. *video streaming*). Dane multimedialne są przesyłane za pomocą zarówno transmisji jednokierunkowych, jak i dwukierunkowych, a ponadto transmisja może być realizowana w czasie rzeczywistym, lub też nie. Dodatkowo, usługi multimedialne różnią się modelem biznesowym, który może wymagać od użytkowników okresowych lub jednorazowych opłat albo udostępniać multimedia za darmo.

Coraz więcej osób rezygnuje z tradycyjnej telewizji na rzecz usług wideo na żądanie. Na przykład, w Stanach Zjednoczonych swój rozkwit przeżywają telewizje dostępne tylko przez Internet, takie jak Netflix, Hulu, czy Amazon, które oferują zarówno licencjonowane treści multimedialne, jak i własne produkcje. Wśród prekursorów serwisów VoD w Polsce są Iplex, Ipla, TVNPlayer i Onet.pl, ale nowe serwisy pojawiają się w bardzo szybkim tempie. W ostatnich latach niebywałą popularność zdobyły przede wszystkim serwisy świadczące swoje usługi bez opłat. Aby zrozumieć skalę tego zjawiska wystarczy spojrzeć na statystyki serwisu YouTube, który jest odwiedzany przez ponad 1 miliard użytkowników miesięcznie, a jego oglądalność to ponad 6 miliardów godzin wideo miesięcznie [1]. Jako wyznacznik sukcesu mogą być uważane również kwoty, za które kupowane są serwisy dystrybucji multimediów. Jako przykład można podać Twitch, darmowy serwis strumieniowania wideo posiadający 55 milionów zarejestrowanych użytkowników, który w 2014 r. został kupiony przez firmę Amazon za 970 milionów dolarów [2]. Należy mieć na uwadze, że różnorodność narzędzi do tworzenia własnych filmów lub muzyki spowodowała, że obecnie źródłami nowych treści multimedialnych nie są jedynie profesjonalne wytwórnie, ale również ciągle rosnąca liczba twórców niezależnych.

Ponieważ cyfrowe multimedia w dzisiejszym świecie posiadają tak dużą wartość, muszą być chronione w odpowiedni sposób. Nielegalne kopiowanie i redystrybucja cyfrowych multimediów prowadzi do łamania praw autorskich. Problem ten, który głównie dotyczy rynku filmowego i muzycznego, codziennie powoduje straty zarówno finansowe, jak i moralne autorów i ich wydawców. Zjawisko nielegalnego udostępniania multimediów nazywane jest piractwem, a osoby łamiące w ten sposób prawa autorskie – piratami. Piractwo stało się bardzo ważnym problemem, który potrzebuje skutecznego rozwiązania gwarantującego, że multimedia będę używane zgodnie z przeznaczeniem i tylko przez uprawnionych do tego użytkowników, którzy uiszczą odpowiednie opłaty. Technologie mające przeciwdziałać używaniu

danych w sposób sprzeczny z wolą ich wydawcy nazywane się mechanizmami zarządzania prawami cyfrowymi DRM (ang. *Digital Rights Management*) [3].

Przede wszystkim konieczne jest, aby zapewnione zostały poufność i integralność przesyłanych danych oraz uwierzytelnienie źródła multimediów, inaczej nazywane ochroną własności danych. Pierwszą linią obrony są metody szyfrowania danych multimedialnych, których celem jest zapewnienie poufności. W tym celu używany jest algorytm szyfrujący, który zamienia oryginalne treści multimedialnych na dane, które przypominają szum lub niosą informację niezrozumiałą dla ludzkich zmysłów. Odszyfrowanie danych jest możliwe tylko przez upoważnionych odbiorców posiadających tajny klucz deszyfrujący. Integralność danych oraz ochrona własności, są realizowane przez metody cyfrowego znaku wodnego (ang. *digital watermarking*), powszechnie zwane watermarkingiem. W metodach watermarkingu, znak wodny (watermark), który jest ciągiem binarnym uwierzytelniającym nadawcę treści multimedialnych, jest osadzany w oryginalnych danych tak, aby był niezauważalny dla człowieka. Następnie, prawowity właściciel danych jest w stanie wydobyć znak wodny w celu weryfikacji integralności lub praw własności chronionych danych.

Niestety, poufność, integralność i ochrona własności danych nie są wystarczające dla systemów dystrybucji multimediów. Po odszyfrowaniu danych, upoważniony odbiorca jest w posiadaniu oryginalnych danych i może je udostępnić nieupoważnionym odbiorcom, łamiąc tym samym prawa autorskie. Na przykład, użytkownik systemu uiszcza opłatę i legalnie odbiera strumień wideo, a następnie rozpowszechnia go innym odbiorcom bez zgody właściciela. Zagadnienie poszukiwania i identyfikacji takich nieuczciwych użytkowników nazywane jest śledzeniem piratów (ang. *traitor tracing*) [3]. W celu rozwiązania tego problemu stosowane są metody cyfrowego odcisku palca (ang. *digital fingerprinting*), powszechnie zwane fingerprintingiem. W metodach fingerprintingu unikatowe ciągi uwierzytelniające zarejestrowanych użytkowników systemu są osadzane w danych multimedialnych, tworząc zbiór danych oznakowanych cyfrowym odciskiem palca (fingerprintem). Cyfrowe odciski palca, podobnie jak cyfrowy znak wodny, są osadzane w sposób niezauważalny dla ludzkich zmysłów, a zatem wszystkie oznakowane kopie wizualnie są identyczne i różnią się tylko ukrytym fingerprintem. W przypadku, gdy kopia z fingerprintem zostanie nielegalnie rozpowszechniona przez nieuczciwego użytkownika, możliwe jest wydobycie z niej osadzonego fingerprintu i na jego podstawie zidentyfikowanie pirata. Łatwo zauważyć, że ochrona praw autorskich za pomocą metod fingerprintingu nie polega na zapobieganiu, ale na przeciwdziałaniu piractwu poprzez identyfikację sprawców.

Należy przyjąć założenie, że nieuczciwi użytkownicy, którzy zamierzają nielegalnie udostępnić multimedia są świadomi obecności fingerprintów w ich kopiach i przeprowadzą różne działania mające na celu ich usunięcie lub przynajmniej silne zniekształcenie. Jednocześnie, piratom zależy na tym, aby usuwając fingerprint nie uszkodzić treści multimedialnej i zachować możliwie najlepszą jej jakość perceptualną. Pirat może uszkodzić fingerprint zawarty w multimediach poprzez np. kompresję, skalowanie obrazu, wycinanie fragmentów obrazu, filtrowanie dolnoprzepustowe sygnału, itp. Jednakże, największe zagrożenie dla metod fingerprintingu stanowią ataki, które są przeprowadzane nie przez jednego, a przez grupę zmówionych piratów. W tych atakach piraci wykorzystują swoje oznakowane kopie tych samych multimediów i na ich podstawie generują piracką kopię, która jest wolna od fingerprintu lub zawiera tak uszkodzony fingerprint, że nie identyfikuje prawdziwych sprawców. Takie

ataki nazywane są atakami zmowy [4, 5, 6, 7, 8] (ang. *collusion attacks*) i są one bardzo trudne do odparcia, ponieważ im więcej jest zmówionych piratów, tym mniejsze jest ryzyko ich identyfikacji. Ponieważ ataki zmowy są zazwyczaj najskuteczniejszą metodą usunięcia osadzonego fingerprintu, zatem istotne jest, aby nowe metody zostały zbadane pod kątem odporności na tego typu ataki przy dużej liczbie zmówionych piratów.

Przy stosowaniu metod fingerprintingu należy pamiętać, że metoda musi działać w środowisku sieciowym. Dla dostawców usług jest ważne, aby koszt świadczenia usługi dla pojedynczego użytkownika był możliwie niski. Zatem, w celu maksymalizowania zysków, dostawcy muszą świadczyć jak największą liczbę usług, przy jednocześnie ograniczonej dostępnej przepustowości. Zatem, konieczne jest wykorzystanie transmisji rozgłoszeniowej typu multicast, która zapewni wysokie wykorzystanie pasma oraz zasobów obliczeniowych dla dystrybucji multimediów. Ponadto, niewątpliwą zaletą systemu dystrybucji multimediów opartego na transmisji multicast jest jego wysoka skalowalność ze względu na rosnącą liczbę odbiorców usługi. Niestety, pomimo tych zalet, realizacja metod fingerprintingu w oparciu o tego typu transmisje jest bardzo trudna. Metody fingerprintingu wymagają, aby każdy z użytkowników otrzymał trochę inną, unikatowo oznakowaną, kopię danych, natomiast transmisja multicast pozwala na przesyłanie dokładnie tej samej treści do wszystkich odbiorców. W celu pokonania tej trudności, w ostatnich latach powstało wiele metod fingerprintingu specjalnie dostosowanych do transmisji multicast.

Pośród metod fingerprintingu wykorzystujących transmisję multicast najbardziej obiecującymi są metody należące do grupy o nazwie łączny fingerprinting i deszyfracja JFD (ang. *Joint Fingerprinting and Decryption*) [9, 10, 11, 12, 13]. Metody JFD łączą w sobie aspekty szyfrowania i fingerprintingu oraz charakteryzują się niezwykłą cechą – osadzają fingeprint po stronie odbiorczej. Strona nadawcza szyfruje dane multimedialne za pomocą grupowego klucza szyfrującego, a następnie przesyła je do wszystkich odbiorców za pomocą transmisji multicast. Każdy odbiorca otrzymuje indywidualny klucz deszyfrujący, różniący się od klucza szyfrującego, który pozwala na odszyfrowanie danych z niewielkimi błędami. Wprowadzone do danych multimedialnych błędy są niezauważalne dla ludzkiego oka i są unikatowe w skali wszystkich użytkowników, a zatem stanowią one ich fingerprinty. Metody JFD są bardzo dobrze skalowalne ze względu na rosnącą liczbę odbiorców z dwóch powodów. Po pierwsze, wykorzystanie transmisji multicast powoduje, że przez sieć przesyłana jest tylko jedna kopia danych zaszyfrowanych, co prowadzi do zmniejszenia wymaganej przepustowości. Po drugie, osadzanie fingerprintu jest realizowane podczas deszyfrowania po stronie odbiorczej, co znacznie zmniejsza wymagania na zasoby obliczeniowe strony nadawczej, ponieważ jedyną operacją wykonywaną przez stronę nadawczą jest szyfrowanie, które odbywa się jednokrotnie, niezależnie od liczby użytkowników systemu.

Jednakże, dotychczas zaproponowane metody JFD posiadają istotne wady, takie jak niewystarczające ukrywanie obrazu w szyfrogramie, niewystarczająca odporność na ataki zmowy lub niewystarczająca odporność na kompresję. Ponadto, większość proponowanych rozwiązań została zbadana tylko dla obrazów w odcieniach szarości, a ich implementacja dla obrazów kolorowych spowodowałaby znaczny wzrost wymagań na zasoby obliczeniowe. Zatem, konieczne jest opracowanie nowych metod JFD, ekonomicznych pod względem wykorzystania zasobów i wolnych od powyższych wad. W tej rozprawie doktorskiej zapropo-

nowano nowe, skuteczne metody łącznego fingerprintingu i deszyfracji dla obrazów kolorowych, które wykorzystują algebrę macierzy oraz algebrę kwaternionów.

Kwaterniony stanowią rozszerzenie ciała liczb zespolonych [14, 15], które powstało w celu opisu obrotu punktów w przestrzeni 3-wymiarowej. Kwaterniony są 4-wymiarowymi liczbami zespolonymi, które składają się z jednej składowej rzeczywistej oraz trzech składowych urojonych. Początkowo, kwaterniony były uważane za dziwny twór, ponieważ nie spełniały reguły przemienności mnożenia. Jednakże, aktualnie kwaterniony znalazły szerokie zastosowanie zarówno w matematyce teoretycznej, jak i stosowanej. Głównym zastosowaniem algebry kwaternionów jest reprezentacja obrotów, odbić i skalowania w przestrzeni 3-wymiarowej, a także łączenia następujących po sobie obrotów. Obecnie, kwaterniony są wykorzystywane w pakietach graficznych DirectX oraz OpenGL, ponieważ przekształcenia w przestrzeni 3-wymiarowej na kwaternionach są wykonywane szybciej niż przy wykorzystaniu innej reprezentacji liczbowej. Oczywiście, obliczanie rotacji w 3-wymiarowej przestrzeni za pomocą 3 współrzędnych lub 3 kątów jest możliwe, ale taka reprezentacja powoduje, że operacje są nieliniowe, mają osobliwości i są trudne do łączenia. Kwaterniony rozwiązują te problemy poprzez zamodelowanie 3-wymiarowych rotacji w 4-wymiarowej przestrzeni. Doskonałą analogią wyjaśniającą ten fenomen jest mapa Świata. Nie jest możliwe przedstawienie powierzchni całej Ziemi na 2-wymiarowej mapie bez zniekształceń kątów lub powierzchni. Jednakże, gdy 2-wymiarowa mapa zostanie owinięta wokół 3-wymiarowej sfery, to wierne odwzorowanie staje się łatwiejsze, a przekształcenia liniowe. Analogicznie, 3-wymiarowe rotacje stają się liniowe, kiedy zastosuje się reprezentację w przestrzeni 4-wymiarowej [15].

W niniejszej rozprawie rozważane są sposoby ochrony obrazów kolorowych. Przy wykorzystaniu znanych reprezentacji kolorów, tj. modelu RGB lub YCbCr, można zdefiniować 3-wymiarową przestrzeń, której poszczególne wymiary stanowią trzy kanały kolorów w przypadku modelu RGB lub kanał luminancji i dwa kanały chrominancji w przypadku modelu YCbCr. Dzięki temu, elementy obrazu można interpretować jako punkty, w tak zdefiniowanej 3-wymiarowej przestrzeni kolorów. W niniejszej pracy zaproponowano metody JFD, w których algorytmy szyfrowania i deszyfrowania można interpretować jako przekształcenia w tej przestrzeni. Ze względu na szczególne właściwości kwaternionów i wygodę ich stosowania w opisie rotacji 3-wymiarowych, zdecydowano, że rachunek kwaternionowy będzie jednym z głównych narzędzi wykorzystywanych przy projektowaniu nowych metod JFD.

1.1. Cel i teza pracy

Teza pracy została sformułowana następująco: **Wykorzystanie kwaternionowej reprezentacji obrazów kolorowych i właściwości rachunku kwaternionowego umożliwiają zrealizowanie systemu łącznego fingerprintingu i deszyfracji odpornego na ataki zmowy.**

Aby wykazać prawdziwość tezy, przyjęto następujące zadania badawcze. Celem pracy jest zaprojektowanie skutecznej metody łącznego fingerprintingu i deszyfracji dla obrazów kolorowych, wykorzystującej rachunek kwaternionowy. Metoda ma obejmować trzy algorytmy: algorytm dla strony nadawczej, który będzie realizować konstrukcję indywidualnych kluczy odbiorców i szyfrowanie obrazu za pomocą klucza grupowego; algorytm dla strony odbiorczej, który ma realizować deszyfrowanie oraz osadzanie cyfrowych odcisków palca (fin-

gerprintów); oraz algorytm detekcji dla strony nadawczej lub zaufanej trzeciej strony, którego przeznaczeniem będzie identyfikacja użytkowników biorących udział w nielegalnej redystrybucji obrazu. Fingerprinty muszą być osadzone w obrazie w sposób niezauważalny dla ludzkiego oka. Projektowana metoda ma być odporna na próby usunięcia osadzanych fingerprintów. Głównym kryterium oceny skuteczności projektowanej metody będzie odporność na ataki zmowy, czyli ataki realizowane przez zorganizowane grupy nieuczciwych użytkowników (piratów).

1.2. Struktura rozprawy

Rozdziały od 1. do 4. stanowią część wstępną rozprawy, która zawiera opis tematyki pracy, przedstawienie problemu badawczego, oraz analizę źródeł literaturowych. Rozdziały od 5. do 10. stanowią główną część rozprawy, która zawiera algorytmy nowych metod łącznego fingerprintingu i deszyfracji dla obrazów kolorowych, szczegółowe omówienie wyników badań ich jakości, oraz opis sprawiedliwego protokołu zarządzania dystrybucją dla metod JFD. Rozdział 11. stanowi zakończenie rozprawy, które zawiera konkluzję o zrealizowaniu celu pracy, podsumowanie informacji zawartych w rozprawie, ocenę wkładu do rozwoju wiedzy oraz perspektywy kontynuacji badań. Rozdział 6. został napisany w celu wyjaśnienia podstawowych operacji na kwaternionach i został zawarty dla zapewnienia kompletności wykładu. Poniżej znajduje się opis zagadnień omawianych w poszczególnych rozdziałach.

W drugim rozdziale rozprawy zostały opisane podstawy teoretyczne cyfrowego znakowania multimediów za pomocą cyfrowego odcisku palca, na przykładzie klasycznego modelu fingerprintingu. Opisano kryteria konstrukcji zbioru ciągów binarnych, które mają pełnić rolę fingerprintów oraz wymagania stawiane metodom cyfrowego znakowania danych. Zdefiniowano także podział metod fingerprintingu ze względu na umiejscowienie w sieci procesu osadzania fingerprintów do danych multimedialnych. Następnie, opisano zagadnienie wykorzystania transmisji multicast do dystrybucji oznakowanych multimediów. Na końcu tego rozdziału, zostały opisane wybrane metody fingerprintingu wykorzystujące transmisję multicast, które nie należą do grupy metod łącznego fingerprintingu i deszyfracji.

W trzecim rozdziale zostały opisane ataki na fingerprinty, z podziałem na ataki realizowane przez pojedynczego użytkownika oraz ataki zmowy, realizowane przez grupę użytkowników. Jako ataki pojedynczego pirata przedstawiono różne rodzaje przetwarzania obrazu oznakowanego, natomiast wśród ataków zmowy zdefiniowano różne ataki liniowe i nieliniowe. Przedstawiono również strategie identyfikacji piratów w przypadku ataków zmowy. Ponadto, wskazano, które ataki wybrano do badania odporności projektowanych metod fingerprintingu, a także, jakie przyjęto kryterium oceny. Ponadto, na końcu rozdziału przedstawiono zalety osadzania fingerprintów we współczynnikach transformaty, czyli w dziedzinie częstotliwości, które ma na celu zwiększenie odporności osadzonych fingerprintów.

Czwarty rozdział został poświęcony koncepcji łącznego fingerprintingu i deszyfracji. Najpierw opisano genezę tych metod oraz ogólne zasady działania. Następnie, przedstawiono kwestie bezpieczeństwa szyfrowania w kontekście JFD. Przedstawiono założenia dotyczące bezpieczeństwa szyfrowania obrazów, które zostały przyjęte podczas projektowania metod JFD. Na końcu zamieszczono przegląd literatury, obejmujący prace, które miały wpływ na powstanie nowych metod JFD. Rozdział czwarty kończy część wstępną rozprawy.

W piątym rozdziale została zaproponowana nowa metoda JFD oparta na iloczynie macierzy. Opis metody składa się z następujących cześci: konstrukcja kluczy, szyfrowanie obrazu, realizowane po stronie nadawczej, łączny fingerprinting i deszyfracja, realizowane po stronie odbiorczej, oraz identyfikacja piratów realizowana przez stronę nadawczą lub zaufaną trzecią stronę. W ostatnim podrozdziale rozpatrzono słaby punkt szyfru Hilla, czyli wrażliwość na atak ze znanym tekstem otwartym, i zaproponowano mechanizm okresowej zmiany klucza, który zapewnia odporność proponowanej metody JFD na ten atak.

W szóstym rozdziale znajduje się wprowadzenie do rachunku kwaternionowego. W rozdziale przedstawiono pojęcie kwaternionu, a następnie zdefiniowano sumę, iloczyn, sprzężenie, odwrotność oraz normę w ciele kwaternionów. Na końcu rozdziału przedstawiono możliwości wykorzystania kwaternionów do opisu transformacji punktów w przestrzeni trójwymiarowej. Opisane operacje zostały wykorzystane w nowych metodach JFD przedstawionych w dwóch kolejnych rozdziałach.

W siódmym rozdziale została zaproponowana nowa metoda JFD oparta na rotacji kwaternionów w trybie wiązania kwaternionów zaszyfrowanych. Analogicznie do przedstawienia metody w rozdziale 5., opis metody składa się z następujących cześci: konstrukcja kluczy, szyfrowanie obrazu, realizowane po stronie nadawczej, łączny fingerprinting i deszyfracja, realizowane po stronie odbiorczej, oraz identyfikacja piratów realizowana przez stronę nadawczą lub zaufaną trzecią stronę.

W ósmym rozdziale została zaproponowana nowa metoda JFD oparta na serii rotacji kwaternionów. Analogicznie do przedstawienia metod w rozdziale 5. i 7., opis metody składa się z następujących cześci: konstrukcja kluczy, szyfrowanie obrazu, realizowane po stronie nadawczej, łączny fingerprinting i deszyfracja, realizowane po stronie odbiorczej, oraz identyfikacja piratów realizowana przez stronę nadawczą lub zaufaną trzecią stronę.

W dziewiątym rozdziale zamieszczono wyniki badań nowych metod JFD dla obrazów kolorowych, zaproponowanych w rozdziałach 5., 7. i 9. Najpierw, przedstawiono parametry przeprowadzonych symulacji oraz wyniki badań niezauważalności osadzonych fingerprintów, odporności na szum, odporności na kompresję, odporności na ataki zmowy, i odporności na ataki zmowy połączone z kompresją. Następnie, porównano nowe metody JFD ze sobą oraz innymi metodami fingerprintingu, znanymi z literatury. Metody zostały porównane w kontekście uzyskanych wyników badań symulacyjnych, a także zgodności z koncepcją JFD oraz trudności implementacji.

W dziesiątym rozdziale poruszono temat sprawiedliwości ochrony, jaką zapewniają metody fingerprintingu. Zwrócono uwagę na fakt, że metody te są projektowane głównie z myślą o ochronie usługodawców przed działaniami nieuczciwych klientów, podczas gdy uczciwi klienci pozostają narażeni na działania nieuczciwych usługodawców. Wyjaśniono, na czym polega problem praw klienta (ang. *customer's rights problem*) oraz problem niepowiązania (ang. *unbinding problem*), a następnie zaproponowano rozwiązanie tych problemów w postaci sprawiedliwego protokołu zarządzania dystrybucją dla metod JFD.

Podsumowanie całej rozprawy, zawierające wyszczególnienie oryginalnych, nowych wyników, konkluzję o zrealizowaniu celu pracy, ocenę osiągniętych rezultatów wraz z krytycznymi uwagami dotyczącymi zawartości pracy oraz perspektywy dalszych badań, przedstawiono w rozdziale jedenastym.

2. OCHRONA MULTIMEDIÓW POPRZEZ FINGERPRINTING

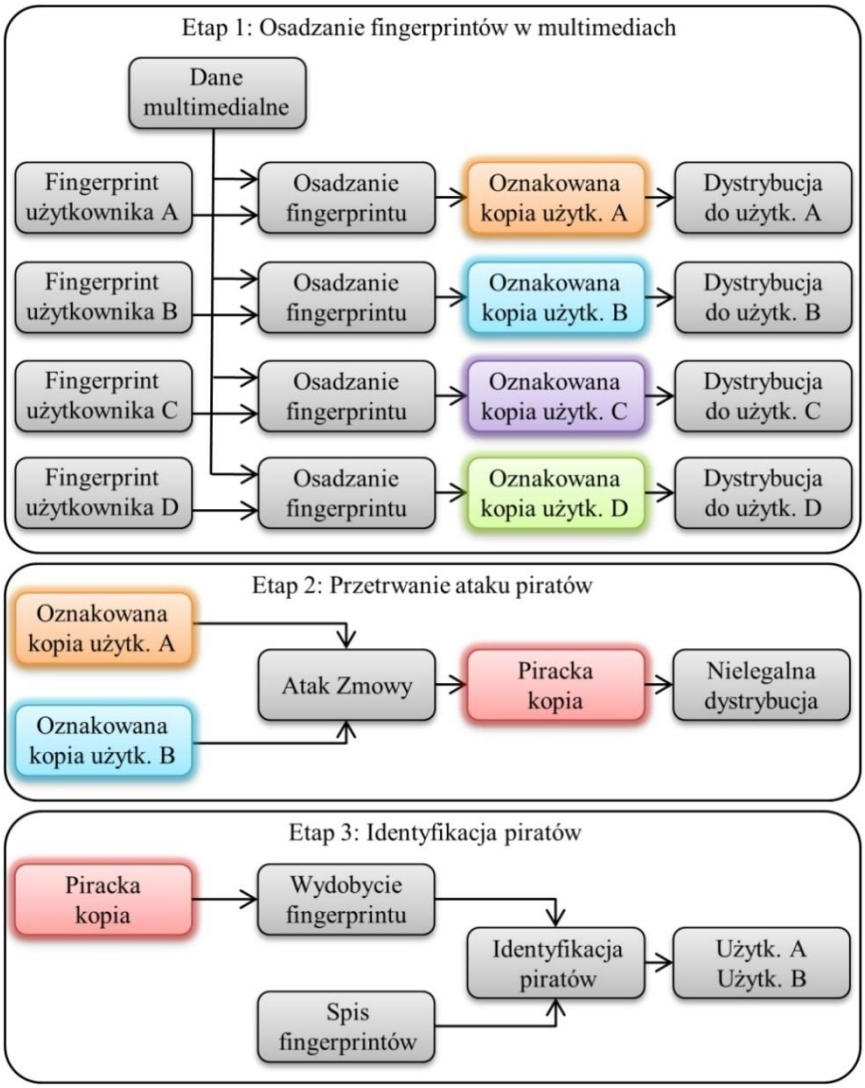
Ochrona treści multimedialnych oraz praw autorskich jest realizowana za pomocą dwóch uzupełniających się technik [3]. Pierwszą techniką jest szyfrowanie, którego przeznaczeniem jest zapewnienie poufności przesyłanych multimediów. Technika ta zapewnia, że tylko użytkownicy posiadający odpowiednie klucze deszyfrujące, będą w stanie odszyfrować przesyłane treści multimedialne. Niestety, szyfrowanie nie jest wystarczającym zabezpieczeniem, gdyż po deszyfracji użytkownik mający dostęp do multimediów może je rozpowszechnić bez zgody autora, łamiąc tym samym prawa autorskie. Drugą techniką jest ochrona śledcza, którą można podzielić na dwa aspekty: cyfrowy znak wodny (ang. *digital watermarking*) oraz cyfrowy odcisk palca (ang. *digital fingerprinting*). Techniki te polegają na ukrywaniu dodatkowych ciągów binarnych w treści multimedialnej. W przypadku watermarkingu, ciągi te nazywane są cyfrowymi znakami wodnymi (watermarkami), które jednoznacznie identyfikują stronę nadawczą. Natomiast, w przypadku fingerprintingu, ciągi te nazywane są cyfrowymi odciskami palca (fingerprintami), które jednoznacznie identyfikują stronę odbiorczą. W obu przypadkach istotne jest, aby osadzane ciągi były perceptualnie niezauważalne. Osadzone dane dają możliwość późniejszej weryfikacji pochodzenia kopii podejrzanej o to, że jest bezprawnie rozpowszechniana. Dane te pozwalają na oskarżenie nieuczciwych użytkowników (piratów) lub poddaniu ich dalszej obserwacji, w celu zgromadzenia dodatkowych dowodów.

Pojęcia watermarking i fingerprinting są pokrewne, ale nie są jednoznaczne. Różnice między watermarkiem, a fingerprintem należy rozumieć tak, jak różnice między ich fizycznymi odpowiednikami. Dystrybutor treści multimedialnych oznacza swoje dane cyfrowym znakiem wodnym, który służy do weryfikacji ich pochodzenia. Odpowiednikiem w świecie fizycznym są banki narodowe, które oznaczają swoje banknoty i papiery wartościowe znakami wodnymi w celu określenia źródła pochodzenia dokumentu oraz uniemożliwienia fałszerstwa. Natomiast, cyfrowe odciski palców są osadzane w multimediach w celu identyfikacji docelowych odbiorców systemu. Odpowiednikiem w świecie fizycznym są odciski palców przestępców, gdyż tak jak obecność prawdziwego odcisku palca w miejscu przestępstwa identyfikuje sprawcę, tak samo obecność fingerprintu w nielegalnej kopii identyfikuje piratów.

W tym rozdziale szczegółowo opisano klasyczne zasady działania metod fingerprintingu, kryteria konstrukcji zbioru fingerprintów, a także wymagania stawiane metodom cyfrowego znakowania danych w różnych zastosowaniach. Następnie, przedstawiono podział metod fingerprintingu ze względu na to, w którym miejscu realizowane jest osadzanie fingerprintów. Poruszono również zagadnienie wykorzystania transmisji multicast w połączeniu z metodami fingerprintingu.

2.1. Klasyczne metody działania fingerprintingu

W tym podrozdziale przedstawiono klasyczny model fingerprintingu. W dalszych rozdziałach zostały opisane metody odbiegające od tego modelu, ale ten model jest wystarczający do wyjaśnienia ogólnej idei systemów fingerprintingu. Działanie systemu fingerprintingu można podzielić na trzy etapy [3]: osadzanie fingerprintów w multimediach, przetrwanie potencjalnego ataku, czyli próby usunięcia fingerprintu z multimediów oraz identyfikacja użytkowników biorących udział w ataku.



Rys. 2.1. Klasyczny model fingerprintingu, opracowanie własne.

Na rysunku 2.1. przedstawiono trzy etapy klasycznego modelu fingerprintingu. W pierwszym etapie, strona nadawcza musi utworzyć zbiór unikatowych ciągów binarnych, które będą pełnić rolę fingerprintów, a następnie osadzić je w multimediach. Dany jest sygnał cyfrowy *x*, w którym mają być osadzone ukryte dane $f^{(u)}$. Sygnał *x* nazywany jest sygnałem hosta lub sygnałem oryginalnym, natomiast dane $f^{(u)}$ są fingerprintem *u*-tego użytkownika, gdzie $u = 1,2,...U$, a *U* to całkowita liczba użytkowników. Konieczne jest, aby wszystkie fingerprinty były jednoznacznie przyporządkowane do poszczególnych użytkowników. W wyniku osadzania danych, otrzymywany jest sygnał $x^{(u)}$, który jest oznakowaną kopią sygnału hosta. Osadzanie danych $f^{(u)}$ musi być przeprowadzone w taki sposób, aby sygnały *x* oraz $x^{(u)}$ były identyczne perceptualnie. W związku z tym, konieczne jest wyskalowanie fingerprintu $f^{(u)}$ tak, aby po dodaniu do sygnału hosta *x* był on niezauważalny dla ludzkiego oka. W tym celu fingerprint $f^{(u)}$ jest mnożony przez współczynnik α:

$$f^{(u)'} = \alpha \cdot f^{(u)}, \tag{2.1}$$

gdzie, $f^{(u)}{}'$ jest wyskalowanym fingerprintem *u*-tego użytkownika, a α jest współczynnikiem siły osadzania fingerprintu. Jako wartość współczynnika α należy wybrać największą możliwą wartość, dla której zachowana zostanie niezauważalność osadzonego fingerprintu. Z tego względu, współczynnik α jest również nazywany współczynnikiem progu zauważalnej różnicy JND (ang. *Just Noticeable Difference*). W niektórych rozwiązaniach [16], współczynnik α może być dopasowywany lokalnie do sygnału hosta i przez to może przyjmować różne wartości dla poszczególnych fragmentów sygnału hosta. W innych rozwiązaniach [13], współczynnik α jest taki sam dla całego sygnału hosta. W klasycznym modelu fingerprintingu, znakowanie multimediów polega na addytywnym osadzaniu wyskalowanych fingerprintów $f^{(u)}{}'$ w sygnale *x*. Osadzanie danych może odbywać się również w specjalnej domenie, np. domenie dyskretnej transformaty kosinusowej DCT (ang. *Discrete Cosine Transform*) [17, 18, 19]. Oznakowana kopia danych $x^{(u)}$ dla *u*-tego użytkownika generowana jest zgodne z wzorem:

$$x^{(u)} = x + f^{(u)}{}' = x + \alpha \cdot f^{(u)}. \tag{2.2}$$

Drugi etap systemu fingerprintingu odbywa się w nieprzyjaznym środowisku. Oznakowana kopia $x^{(u)}$ może ulec zniekształceniom w wyniku kompresji, filtracji, lub też w wyniku ataków piratów. Użytkownicy, którzy mają zamiar złamania praw autorskich, mogą przeprowadzić atak mający na celu wygenerowanie sygnału pozbawionego fingerprintu, który wskazywałby sprawców przestępstwa. Tak zmieniony sygnał $x_{\text{test}}^{(u)}$ nazywany jest sygnałem testowym, a różnicę między sygnałami $x^{(u)}$ oraz $x_{\text{test}}^{(u)}$ można łącznie opisać jako addytywne zakłócenie n_{A}. Sygnał testowy pochodzący od *u*-tego użytkownika można przedstawić wzorem:

$$x_{\text{test}}^{(u)} = x + f^{(u)}{}' + n_{\text{A}} = x + \alpha \cdot f^{(u)} + n_{\text{A}}. \tag{2.3}$$

Ważne jest, aby osadzony fingerprint przetrwał manipulacje piratów i był obecny w sygnale testowym $x_{\text{test}}^{(u)}$, który jest kopią nielegalnie rozpowszechnioną przez *u*-tego użytkownika.

Trzeci etap działania systemu fingerprintingu rozpoczyna się, gdy piracka kopia zostaje przechwycona przez stronę nadawczą lub zaufaną stronę trzecią. Sygnał testowy $x_{\text{test}}^{(u)}$ jest podawany na wejście detektora, który ma za zadanie wydobyć z niego fingerprint $f^{(u)}$. Algorytmy wydobycia fingerprintu mogą się bardzo różnić w zależności od stosowanych metod fingerprintingu, ale wyróżnia się ich dwa podstawowe rodzaje: detekcja ślepa (ang. *blind detection*), nazywana niekoherentną oraz detekcja nie-ślepa (ang. *nonblind detection*), nazywana koherentną. W literaturze, pojęcia detekcja fingerprintu i wydobycie (ekstrakcja) fingerprintu są używane zamiennie. Na końcu trzeciego etapu, wydobyty fingerprint jest porównywany z fingerprintami wszystkich zarejestrowanych użytkowników, w celu identyfikacji tożsamości nieuczciwych użytkowników, którzy złamali prawa autorskie. Zazwyczaj odbywa się to poprzez analizę korelacji między fingerprintem wydobytym z sygnału testowego, a fingerprintami przypisanymi do poszczególnych użytkowników. W tym celu zazwyczaj wykorzystuje się funkcję korelacji albo współczynnik korelacji.

Detekcja niekoherentna (ślepa) polega na tym, że wydobycie fingerprintu realizowane jest wyłącznie na podstawie sygnału testowego, tj. sygnał hosta nie jest znany. W tym przypadku sygnał hosta jest traktowany jako zakłócenie, co bardzo utrudnia detekcję, ponieważ sygnał hosta ma znacznie wyższy poziom niż osadzony fingerprint. Dla detekcji niekoherentnej, łączne zakłócenie n_{B} oraz sygnał testowy $x_{\mathrm{test}}^{(u)}$ mają następującą postać:

$$n_{\mathrm{B}} = n_{\mathrm{A}} + x\,, \tag{2.4}$$

$$x_{\mathrm{test}}^{(u)} = f^{(u)\prime} + n_{\mathrm{B}} = \alpha \cdot f^{(u)} + n_{\mathrm{B}}\,. \tag{2.5}$$

Z związku z tym, dla detekcji niekoherentnej stosunek poziomu fingerprintu do szumu FNR (ang. *Fingerprint-to-Noise Ratio*) jest zdefiniowany następującym wzorem:

$$\mathrm{FNR} = 10\log\frac{P_f}{P_n + P_x} \tag{2.6}$$

gdzie P_f to średnia moc osadzonego fingerprintu $f^{(u)}$, P_n to średnia moc zakłóceń z powstałych w wyniku ataku, a P_x to średnia moc sygnału hosta *x*.

Detekcja koherentna (nie-ślepa) polega na tym, że przy wydobyciu fingerprintu z sygnału testowego jest znany jest również sygnał hosta. Odjęcie sygnału hosta od sygnału testowego, zdefiniowanego wzorem (2.3), znacznie ułatwia wydobycie fingerprintu, ponieważ sygnał hosta nie jest traktowany jak zakłócenie. W związku z tym, dla detekcji koherentnej stosunek poziomu fingerprintu do szumu FNR jest większy niż dla detekcji niekoherentnej i wynosi:

$$\mathrm{FNR} = 10\log\frac{P_f}{P_n}, \tag{2.7}$$

gdzie P_f to średnia moc osadzonego fingerprintu $f^{(u)}$, a P_n to średnia moc zakłóceń z powstałych w wyniku ataku. Oznacza to, że detekcja koherentna jest łatwiejsza, i zazwyczaj skuteczniejsza, niż detekcja niekoherentna. Wadą detekcji koherentnej jest to, że konieczna jest znajomość sygnału hosta podczas ekstrakcji fingerprintu, a przechowywanie nieoznakowanych sygnałów wymaga znacznych zasobów.

2.2. Kryteria konstrukcji zbioru fingerprintów

W ogólności, zbiór dowolnych ciągów binarnych może być wykorzystany jako zbiór fingerprintów użytkowników. Zazwyczaj wykorzystuje się w tym celu ciągi realizacji zmiennych losowych, niezależnych statystycznie, o jednakowych rozkładach prawdopodobieństwa, np. o rozkładzie Gaussowskim [20, 21, 22]. Zaletą takich zbiorów fingerprintów jest łatwość wytwarzania oraz ich duża liczebność. Jednakże, jak wyjaśniono w poprzednim podrozdziale,

identyfikacja piratów zazwyczaj jest realizowana w oparciu o badanie korelacji fingerprintu wydobytego z pirackiej kopii oraz fingerprintów poszczególnych użytkowników. Oznacza to, że w celu zwiększenia skuteczności wykrywania piratów należy stosować fingerprinty posiadające odpowiednie właściwości korelacyjne. Pożądane jest, aby ciągi używane jako fingerprinty posiadały współczynnik autokorelacji (ang. *auto-correlation coefficient*) bliski 1 oraz współczynniki korelacji skrośnej (ang. *cross-correlation coefficient*) bliskie 0. Wysoka wartość współczynnika autokorelacji powoduje, że dany fingerprint jest łatwo rozróżnialny przez detektor korelacyjny, co zwiększa prawdopodobieństwo poprawnej identyfikacji fingerprintu pirata. Natomiast, niskie wartości współczynników korelacji skrośnej powodują, że po ataku fingerprint pirata nie upodobni się silnie do fingerprintu innego użytkownika, co zmniejsza prawdopodobieństwo oskarżenia niewinnego użytkownika. Niestety, przyjęcie takich dodatkowych wymagań powoduje znaczną redukcję liczebności zbioru dostępnych fingerprintów.

Pożądane właściwości korelacyjne oraz liczba dostępnych ciągów są głównymi wymaganiami na konstrukcję zbioru fingerprintów. Poza nimi można wymienić następujące, drugorzędne wymagania: fingerprinty powinny przypominać szum, oraz powinny być łatwe do wygenerowania. Analizując powyższy zbiór wymagań można zauważyć, że identyczne wymagania są stawiane ciągom rozpraszającym stosowanym w metodzie wielodostępu CDMA, w której wykorzystywane są *m*-ciągi [23, 24, 25], ciągi Golda [23, 24, 25], ciągi Kasami [26, 27, 24, 25, 28] oraz ciągi Hadamarda [24, 25]. Ponieważ ciągi rozpraszające, opracowane dla wielodostępu CDMA, charakteryzują się pożądanymi właściwościami, zatem naturalne jest ich wykorzystanie w roli fingerprintów.

Zbiory ciągów Golda oraz ciągów Kasami charakteryzują się większą liczebnością zbioru niż *m*-ciągi, ale ważniejszą cechą są ich niskie i regularne wartości korelacji skrośnej. Zarówno w przypadku ciągów Golda, jak i ciągów Kasami, funkcja korelacji skrośnej dowolnych dwóch ciągów ze zbioru jest trójwartościowa, a wartości jakie przyjmuje są zależne od długości ciągu M [23, 24, 25]. Ponadto, funkcja autokorelacji, poza punktem szczytowym w 0, również jest trójwartościowa i przyjmuje takie same wartości. Długość *m*-ciągów, ciągów Golda i ciągów Kasami jest nieparzysta i wynosi $M = 2^m - 1$, gdzie *m* jest długością rejestru przesuwnego LSFR (ang. *Linear Feedback Shift Register*). Może to być niekorzystne w przypadku, gdy długość osadzanego ciągu ma być potęgą liczby 2 lub przynajmniej ma być parzysta. W takim przypadku, na końcu ciągu może być dodany jest jeszcze jeden bit o stałej wartości. Innym rozwiązaniem jest dodanie bitu o wartości „1” w najdłuższej serii bitów o wartości „0” (lub „–1” w przypadku ciągów bipolarnych), tworząc tzw. ciągi rozszerzone. Natomiast, ciągi Hadamarda są względem siebie ortogonalne, czyli współczynnik korelacji skrośnej dowolnej pary ciągów jest równy 0, a współczynnik autokorelacji dowolnego ciągu jest równy 1. Niestety, ta właściwość jest zachowana tylko przy doskonałej synchronizacji między badanymi ciągami. Jeżeli badana para ciągów nie jest zsynchronizowana, to ortogonalność nie zostanie zachowana, a wartość korelacji skrośnej dla niektórych par ciągów może przyjmować duże wartości. W przypadku addytywnego osadzania fingerprintów, np. w klasycznym modelu przedstawionym w rozdziale 2.1., zazwyczaj możliwe jest dokładne wydobycie fingerprintu z pirackiej kopii i zsynchronizowanie go z fingerprintami użytkowników.

We wstępnych badaniach przeprowadzonych w pracach [29, 30], sprawdzono wykorzystywanie powyższych rodzajów ciągów rozpraszających w roli fingerprintów. Dla nieaddytywnego osadzania fingerprintów, które jest wykorzystywane w nowych metodach JFD, za-

proponowanych w niniejszej rozprawie, najlepsze rezultaty otrzymano przy użyciu ciągów Golda oraz ciągów Kasami. Ciągi Hadamarda, pomimo ich ortogonalności, nie dały najlepszych rezultatów, ponieważ w proponowanych metodach fingerprint nie jest wydobywany w swojej oryginalnej formie, ale jest w pewien sposób związany z chronionym obrazem. Taka zależność powoduje, że synchronizacja fingerprintów jest bardzo trudna do osiągnięcia. Podczas badań nowych metod JFD, w roli fingerprintów użyto ciągów Golda ze względu na ich korzystne właściwości korelacyjne, dużą liczność oraz łatwość generowania zbioru.

Całkowicie inne podejście do konstrukcji zbioru fingerprintów reprezentują kody Tardos [31] oraz kody Boneh-Shaw [32]. W tym podejściu, występuje pojęcie fingerprintów *C*-bezpiecznych (ang. *C-secure collusion-resistant codes*), czyli fingerprintów odpornych na ataki zmowy co najwyżej *C* piratów z prawdopodobieństwem sukcesu większym niż $1-\varepsilon$, gdzie jako sukces należy rozumieć sytuację, w której zbiór użytkowników, wskazany przez algorytm detekcji, zawiera się w zbiorze *C* zmawiających się piratów. Podstawowym założeniem dla tego typu fingerprintów jest tzw. założenie znakowania MA (ang. *Marking Assumption*) [32, 33], które oznacza, że zmawiający się piraci mogą modyfikować tylko te elementy danych, które są w ich kopiach różne. Elementy te są nazywane pozycjami wykrywalnymi (ang. *detectable positions*), a elementy identyczne w różnych kopiach piratów – pozycjami niewykrywalnymi (ang. *undetectable positions*). Ponadto, zakłada się, że jeden element chronionych danych (jedna pozycja) zawiera jeden element fingerprintu. Wówczas, identyczne elementy dwóch fingerprintów są niewykrywalne dla piratów i nie mogą być zmodyfikowane.

Pomimo swoich interesujących właściwości, fingerprinty odporne na ataki zmowy nie zostały wykorzystane w nowych metodach JFD zaprezentowanych w tej rozprawie. Pierwszym powodem jest to, że założenie znakowania MA (ang. *Marking Assumption*) nie jest dla nich spełnione. Dzięki zastosowaniu nieaddytywnego osadzania fingerprintów, jeden element (jedna pozycja) chronionego obrazu zawiera więcej niż jeden element fingerprintu. Drugim powodem jest to, że w nowych metodach fingerprint nie jest wydobywany w swojej oryginalnej formie, ale jest w pewien sposób związany z chronionym obrazem, co uniemożliwia zastosowanie algorytmu detekcji wymaganego przez kody Boneh-Shaw lub kody Tardos.

2.3. Wymagania dla metod fingerprintingu

Istnieje sześć głównych wymagań stawianych metodom cyfrowego znakowania danych multimedialnych, które można podzielić na trzy wymagania związane z właściwościami osadzonych ciągów binarnych [3, 34] oraz trzy wymagania związane z ogólną sprawnością systemu [35]. Wymaganiami na algorytm osadzania ciągów są: niezauważalność, odporność na usunięcie oraz długość osadzanych ciągów binarnych. Natomiast, wymaganiami na sprawność systemu są: wydajność konstrukcji oznakowanych danych, wydajność dystrybucji oznakowanych danych oraz wydajność detekcji osadzonych fingerprintów.

Pierwszym wymaganiem jest niezauważalność osadzonych danych. Osadzanie dodatkowych danych w multimediach powoduje pewne zmiany w sygnale hosta, tymczasem stronie nadawczej zależy na tym, aby jakość perceptualna treści multimedialnej pozostała niezmieniona lub była przynajmniej bardzo zbliżona do oryginalnej. Zatem konieczne jest utrzymywanie poziomu zmian w multimediach poniżej poziomu zmian zauważalnych dla ludzkich zmysłów. Drugim wymaganiem jest odporność na usunięcie osadzonych danych.

Odporność na usunięcie jest bardzo ważna, gdyż fingerprinty nie spełnią swojego przeznaczenia, jeżeli nie przetrwają ataków przeprowadzanych przez piratów. Na odporność osadzonych ciągów ma wpływ zarówno sposób osadzania, jak i rodzaj tych ciągów. Zwiększenie odporności na ataki zazwyczaj powoduje, że osadzone dane są bardziej widoczne, z kolei lepsze ukrycie osadzonych danych powoduje, że ich usunięcie jest łatwiejsze. Zatem, konieczne jest osiągnięcie kompromisu między niezauważalnością, a odpornością osadzonych danych. Kolejnym wymaganiem jest długość osadzanych ciągów, która jest istotna, gdy osadzane ciągi nie tylko identyfikują jedną ze stron, ale również przenoszą dodatkowe informacje.

Wydajność konstrukcji dotyczy złożoności obliczeniowej procesu generowania oznakowanych kopii multimediów. Jeżeli złożoność obliczeniowa jest wysoka i szybko rośnie wraz z liczbą klientów, to wydajność konstrukcji oznakowanych danych jest niska i *vice versa*. Ponieważ ten rodzaj wydajności odnosi się do zasobów obliczeniowych dostępnych stronie nadawczej, zatem jest on bardzo istotny z punktu widzenia minimalizowania kosztów utrzymania systemu. Podobnie, wydajność detekcji dotyczy złożoności obliczeniowej procesu wydobycia fingerprintu z oznakowanych danych i identyfikacji piratów. Jeżeli złożoność obliczeniowa jest wysoka, to oznacza to, że wydajność detekcji fingerprintów jest niska i *vice versa*. Ostatnim wymaganiem jest wydajność dystrybucji oznakowanych danych. Ten rodzaj wydajności dotyczy szerokości pasma, które jest używane podczas transmisji wszystkich oznakowanych kopii multimediów przez sieć. Im wyższej przepustowości wymaga transmisja, tym niższa jest wydajność dystrybucji oznakowanych danych. Ponieważ ten rodzaj wydajności odnosi się bezpośrednio do zasobów sieciowych, dostępnych stronie nadawczej, zatem jest on bardzo istotny z punktu widzenia minimalizowania kosztów utrzymania systemu.

Zaprojektowanie metody cyfrowego znakowania danych spełniającej w równym stopniu wszystkie sześć wymagań byłoby niezwykle trudne. Na szczęście, nie jest to konieczne, gdyż dla różnych zastosowań poszczególne wymagania mają różną wagę. Istnieje pięć głównych zastosowań cyfrowego znakowania danych [3]: ochrona własności, wykrywanie manipulacji danych, kontrola dostępu do multimediów, adnotacje oraz cyfrowy odcisk palca, który jest zastosowaniem rozpatrywanym w tej pracy. W przypadku ochrony własności, osadzane dane stanowią znak wodny wydawcy, a zawarta w nim informacja ma uniemożliwić fałszowanie tożsamości źródła pochodzenia oznakowanej kopii. Wykrywanie manipulacji polega na tym, że w multimediach osadzane są długie ciągi bitów, które po późniejszej analizie mają na celu sprawdzenie integralności oznakowanej kopii. Kontrola dostępu jest związana z urządzeniami do odtwarzania multimediów, a osadzane dane zawierają informacje na temat blokady dostępu do multimediów lub braku możliwości ich powielania. W przypadku adnotacji, osadzane dane nie przyczyniają się bezpośrednio do ochrony praw autorskich, natomiast stanowią one narzędzie do przesyłania dodatkowych informacji, np. kluczy, lub informacji o autorze. Natomiast cyfrowy odcisk palca, jak opisano to na początku tego rozdziału, polega na tym, że w multimediach są osadzane dodatkowe dane w celu późniejszej identyfikacji nieuczciwych użytkowników, którzy łamią prawa autorskie udostępniając pirackie kopie multimediów.

W tabeli 2.1. zawarto priorytety odzwierciedlające istotność poszczególnych wymagań dla różnych zastosowań cyfrowego znakowania danych multimedialnych. W dalszej części rozprawy, cyfrowe znakowanie danych będzie opisywane tylko w kontekście cyfrowego odcisku palca. Wydajność konstrukcji i dystrybucji mają wyższy priorytet niż wydajność detekcji, ponieważ mają one bezpośredni wpływ na koszt i jakość usługi dystrybucji multimediów.

Niezauważalność osadzonych fingerprintów również ma wysoki priorytet, gdyż ma bezpośredni wpływ na jakość oferowanych treści multimedialnych. Natomiast, najważniejszym wymaganiem jest odporność na ataki, gdyż usunięte lub silnie uszkodzone fingerprinty nie spełnią swojego przeznaczenia, czyli nie zidentyfikują nieuczciwych użytkowników.

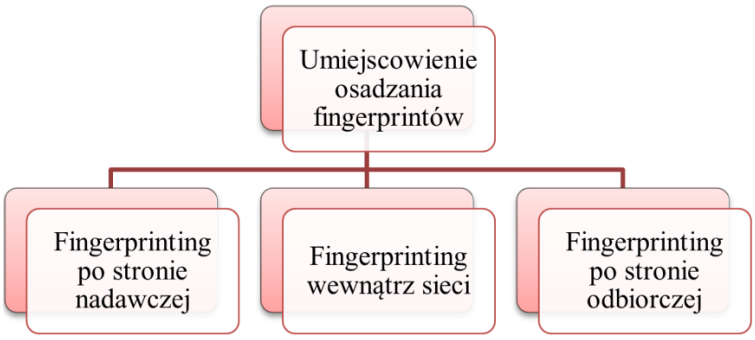
Tab. 2.1. Priorytety odzwierciedlające istotność poszczególnych wymagań dla różnych zastosowań cyfrowego znakowania danych multimedialnych.

		Wymagania					
		Niezauważalność	Odporność na ataki	Długość ciągu	Wydajność konstrukcji	Wydajność detekcji	Wydajność dystrybucji
Zastosowania	Ochrona własności	średni	wysoki	średni	niski	niski	średni
	Wykrywanie manipulacji	wysoki	wysoki	wysoki	średni	niski	średni
	Kontrola dostępu	niski	wysoki	niski	niski	wysoki	średni
	Adnotacje	wysoki	wysoki	wysoki	średni	średni	średni
	Odcisk palca	**wysoki**	**wysoki**	**niski**	**wysoki**	**niski**	**wysoki**

2.4. Umiejscowienie fingerprintingu i wykorzystanie transmisji multicast

W ostatnich latach, zaproponowano metody fingerprintingu, które odbiegają od klasycznego modelu opisanego w podrozdziale 2.1. Przede wszystkim, należy zwrócić uwagę na to, że metody fingerprintingu mogą różnić się miejscem, w którym jest realizowane osadzanie fingerprintów. W konsekwencji, możliwy jest następujący podział na trzy grupy [36, 37, 38]: fingerprinting realizowany po stronie nadawczej, fingerprinting realizowany wewnątrz sieci, oraz fingerprinting realizowany po stronie odbiorczej, co zostało pokazane na rysunku 2.2.

Fingerprinting realizowany po stronie nadawczej odpowiada klasycznemu modelowi. W tej grupie metod, cyfrowe odciski palca są osadzane w danych multimedialnych niezależnie dla każdego odbiorcy. Każda z tych oznakowanych kopii jest oddzielnie szyfrowana i wysyłana niezależnie do poszczególnych użytkowników za pomocą wielu transmisji typu unicast. Po stronie odbiorców przeprowadzana jest tylko deszyfracja, po której oznakowane dane multimedialne są dostępne do użycia. Ten scenariusz realizacji fingerprintingu ma kilka wad. Po pierwsze, liczba obliczeń wykonywanych przez stronę nadawczą rośnie liniowo wraz z liczbą użytkowników, ponieważ osadzanie fingerprintu i szyfrowanie jest realizowane niezależnie dla każdego odbiorcy. Po drugie, dostępna przepustowość jest nieefektywnie wykorzystywana, ponieważ przez sieć jest przesyłanych wiele strumieni danych, które wizualnie niewiele się różnią, gdyż fingerprinty są niezauważalne. Powoduje to, że łączne pasmo wymagane do realizacji usługi dla wszystkich klientów i liczba utrzymywanych połączeń również rośnie liniowo wraz z liczbą użytkowników. Przykładami fingerprintingu realizowanego po stronie nadawczej są: metoda z rozpraszaniem widma I.J. Cox'a i innych [20], addytywny



Rys. 2.2. Podział metod fingerprintingu ze względu na umiejscowienie procesu osadzania fingerprintów do danych multimedialnych, opracowanie własne.

model Chen i Lian [39] oraz hierarchiczny fingerprinting zaproponowany przez M. Kuribayashi'ego [40, 41]. Szyfrowanie i fingerprinting mogą być połączone w celu zmniejszenia liczby obliczeń i takie rozwiązania nazywane są łącznym fingerprintingiem i szyfrowaniem JFE (ang. *Joint Fingerprinting and Encryption*) [42, 36] lub łącznym watermarkingiem i szyfrowaniem JWE (ang. *Joint Watermarking and Encryption*) [43, 44, 45].

W Internecie realizowane są usługi dystrybucji multimediów, w których liczba klientów może być bardzo duża. Dla dostawców tych usług jest ważne, aby koszt transmisji pojedynczej kopii był możliwie niski. Zatem, poszukiwane jest rozwiązanie, które zapewni optymalne wykorzystanie pasma oraz zasobów obliczeniowych. Fingerprinting realizowany po stronie nadawczej jest rozwiązaniem bardzo nieefektywnym ze względu na wykorzystanie połączeń typu unicast. Natomiast, znacznie lepsze wykorzystanie zasobów sieciowych i obliczeniowych można osiągnąć dzięki wykorzystaniu transmisji typu multicast [46, 47], które doskonale się nadają do dystrybucji multimediów na szeroką skalę z powodu swojej wysokiej skalowalności. W dystrybucji multicastowej, nadawca wysyła pakiety do grupy multicastowej składającej się z dowolnej liczby odbiorców. Odbiorcy są identyfikowani przez adres grupy multicastowej. Strumień pakietów jest przesyłany przez sieć do wszystkich odbiorców w optymalny sposób, tzn. pakiety są duplikowane tylko w sytuacji, kiedy przesyłany strumień musi być kierowany różnymi ścieżkami routingu. W rezultacie, między każdą parą węzłów w sieci przesyłana jest tylko jedna kopia danych multimedialnych, niezależnie od liczby docelowych odbiorców. dzięki czemu zapotrzebowanie na przepustowość jest minimalne. Niestety, nie jest możliwe bezpośrednie stosowanie transmisji multicast dla dowolnej metody fingerprintingu, ponieważ cele multicastu i fingerprintingu są niejako przeciwne. Metody fingerprintingu wymagają, aby każdy z użytkowników otrzymał inną kopię danych, natomiast transmisja multicast pozwala na przesyłanie dokładnie tej samej treści do wszystkich odbiorców. Z tego powodu, konieczne było stworzenie nowych metod, realizujących fingerprinting poza stronie nadawczą i dzięki temu pozwalających na wykorzystanie transmisji multicast.

Transmisje typu multicast mogą być wykorzystane w metodach realizujących fingerprinting wewnątrz sieci. W tym scenariuszu, metoda wykorzystuje urządzenia sieciowe, które są zdolne do osadzania własnych watermarków w przesyłanych strumieniach danych multimedialnych. Strona nadawcza szyfruje dane i wysyła je do wszystkich odbiorców za pomocą transmisji multicast. Kolejne fragmenty fingerprintu użytkownika są stopniowo osadzane w strumieniu danych multimedialnych podczas przesyłania ich przez sieć. W rezultacie, fingerprint odbiorcy jest kompozycją watermarków urządzeń sieciowych, przez które przesłano

multimedia, oraz odzwierciedla on lokalizację odbiorcy w sieci. Taka koncepcja zapewnia wysoką skalowalność, jednakże jest to rozwiązanie nieakceptowalne w Internecie, ze względu na wymagany specjalistyczny sprzęt sieciowy, który byłby w stanie odpowiednio interpretować przesyłane pakiety i osadzać w nich ciągi znakujące. Wiąże się to z dodatkowymi kosztami wdrażania tego rozwiązania oraz problemem umiejscowienia tych elementów sieciowych w istniejącej infrastrukturze. Ponadto, urządzenia te musiałyby być również zabezpieczone przed jakąkolwiek ingerencją osób trzecich. Przykładem fingerprintingu realizowanego wewnątrz sieci jest metoda WHIM zaproponowana przez M. Ammar i P. Judge [48].

Fingerprinting realizowany po stronie odbiorczej pierwszy raz został wprowadzony w cyfrowych kinach [49]. Jedyną operacją wykonywaną przez stronę nadawczą jest szyfrowanie, które wykonywane jest jednorazowo niezależnie od liczby odbiorców. Następnie, zaszyfrowane dane są wysyłane do wszystkich użytkowników za pomocą transmisji multicast. Po stronie odbiorczej, musi nastąpić odszyfrowanie odebranych danych, a potem osadzenie w nich fingerprintu odbiorcy. Zaletą tego scenariusza fingerprintingu jest wysoka skalowalność, ponieważ złożoność obliczeniowa algorytmu po stronie nadawczej oraz wymagana przepustowość są niezależne od liczby odbiorców. Niestety, istnieją dwie poważne wady takiego podejścia. Po pierwsze, złożoność obliczeniowa algorytmu po stronie odbiorczej jest większa niż w klasycznym modelu, ponieważ odbiornik nie tylko musi odszyfrować dane, ale również osadzić w nich fingerprint. Po drugie, między deszyfracją, a osadzeniem fingerprintu, odbiorca ma dostęp do sygnału hosta, co może być potencjalnym punktem ataku. Jednakże, istnieje grupa metod łącznego fingerprintingu i deszyfracji JFD (ang. *Joint Fingerprinting and Decryption*) [9, 10, 11, 12, 13], które nie posiadają tych wad. W metodach tych, proces deszyfrowania multimediów oraz proces osadzania w nich fingerprintu są połączone w jedną operację. Dzięki temu, złożoność obliczeniowa algorytmu po stronie odbiorczej nie ulega zwiększeniu oraz odbiornik nie ma dostępu do sygnału hosta, gdyż fingerprint jest osadzany w trakcie deszyfrowania. Rozdział 4. zawiera opis koncepcji łącznego fingerprintingu i deszyfracji, a także przegląd literatury metod JFD.

2.5. Podsumowanie

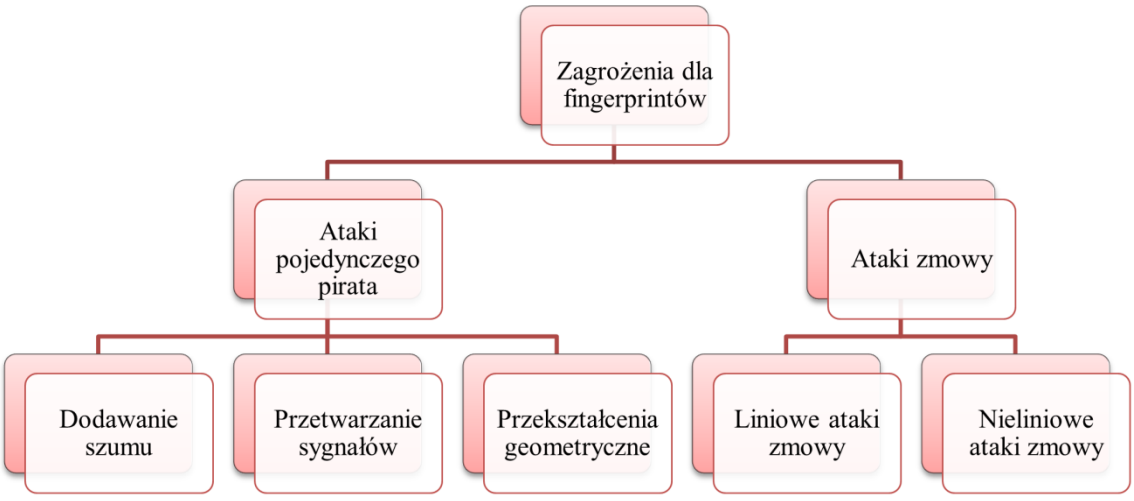
W rozdziale przedstawiona została idea ochrony praw autorskich treści multimedialnych z wykorzystaniem technik cyfrowego znakowania danych. Opisano klasyczny model fingerprintingu i przedstawiono ogólny model matematyczny osadzania i detekcji odcisków palców. Stwierdzono, że głównymi wymaganiami dla ciągów binarnych pełniących rolę fingerprintów jest wysoka wartość współczynnika autokorelacji oraz niskie wartości współczynników korelacji skrośnej. Zaproponowano użycie ciągów rozpraszających stosowanych w CDMA jako fingerprintów użytkowników systemu dystrybucji multimediów. Określono zbiór wymagań dla metod cyfrowego znakowania danych, który zawiera: niezauważalność osadzonych ciągów, odporność na ataki, długość ciągów, wydajność konstrukcji, wydajność detekcji i wydajność dystrybucji oznakowanych danych. Określono również priorytety powyższych wymagań w różnych zastosowaniach. W ostatniej części rozdziału, przedstawiono podział metod fingerprintingu ze względu na umiejscowienie operacji osadzania fingerprintów. Przedstawiono problem efektywnej dystrybucji multimediów do dużej liczby użytkowników i wyjaśniono potrzebę stosowania metod wykorzystujących transmisję typu multicast.

3. ZAGROŻENIA DLA OSADZONYCH FINGERPRINTÓW

Cyfrowy odcisk palca ma za zadanie wskazać nieuczciwego użytkownika, który nielegalnie rozpowszechnił badaną kopię danej treści multimedialnej. W idealnym scenariuszu identyfikacji pirata, multimedia z fingerprintem są nielegalnie rozpowszechniane bez wcześniejszych modyfikacji przeprowadzonych przez pirata. W takim przypadku, algorytm detekcji, będący częścią metody fingerprintingu, powinien bezbłędnie ujawnić, który użytkownik systemu jest piratem. Taki idealny scenariusz jest oczywiście daleki od rzeczywistości, w której pirat jest świadomy istnienia fingerprintingu i podejmuje działania mające zapobiec identyfikacji jego tożsamości. Pirat nie ma dostępu do nieoznakowanej kopii multimediów (sygnału hosta) oraz, w ogólności, nie posiada żadnej wiedzy na temat osadzonego fingerprintu, tj. nie zna długości fingerprintu, struktury fingerprintu, sposobu jego osadzania i lokalizacji w multimediach. Zatem, działania podejmowane przez pirata mają charakter destrukcyjny, tzn. są to modyfikacje obrazu wprowadzające dodatkowe zmiany w oznakowaną kopię multimediów, których celem jest całkowite lub częściowe zniekształcenie niewidocznego fingerprintu. Takie próby usunięcia fingerprintu nazywane są atakiem.

Ponieważ fingerprint jest osadzony w taki sposób, że jest niezauważalny dla ludzkiego oka, to jego zakłócenie poprzez silne zniekształcenia obrazu jest stosunkowo łatwe do wykonania. Jednakże, należy mieć na uwadze fakt, że pirat chce rozpowszechnić multimedia w celu osiągnięcia pewnych korzyści, zazwyczaj majątkowych. Zatem, piracka kopia multimediów musi posiadać walory wizualne zbliżone do walorów kopii oryginalnej i kopii oznakowanej fingerprintem. Oznacza to, że modyfikacje wprowadzane w procesie ataku nie mogą znacząco pogarszać jakości perceptualnej multimediów.

Podział zagrożeń dla osadzonych fingerprintów został przedstawiony na rysunku 3.1. Zagrożenia te można podzielić na dwie grupy: ataki realizowane przez jednego pirata oraz ataki zmowy, które są realizowane przez zorganizowane grupy nieuczciwych użytkowników. Do ataków realizowanych przez pojedynczego pirata należą dodawanie szumu, standardowe metody przetwarzania sygnałów cyfrowych i przekształcenia geometryczne obrazu. Natomiast, pośród ataków zmowy można wyszczególnić ataki zmowy liniowe oraz nieliniowe.



Rys. 3.1. Podział zagrożeń dla osadzonych fingerprinty, opracowanie własne.

W tym rozdziale, szczegółowo opisano każdy rodzaj ataku, a na następnie przedstawiono różne strategie identyfikacji piratów w przypadku ataków zmowy. Wskazano również, które ataki wybrano do badania odporności projektowanych metod fingerprintingu, a także, jakie przyjęto kryterium ich oceny. Na końcu rozdziału przedstawiono zalety osadzania fingerprintów we współczynnikach transformaty, czyli w dziedzinie częstotliwości, które ma na celu zwiększenie odporności osadzonych fingerprintów.

3.1. Ataki realizowane przez pojedynczego pirata

Pojedynczy, nieuczciwy użytkownik może spróbować usunąć fingerprint z oznakowanej kopii multimediów wykorzystując dodawanie szumu, standardowe metody przetwarzania sygnałów cyfrowych, lub przekształcenie geometryczne obrazu [3, 20, 50]. W tym rozdziale omówione zostały takie zagrożenia ze strony pojedynczego pirata.

Dodawanie szumu do obrazu oznakowanego jest stosunkowo prostym atakiem, który może przeprowadzić pojedynczy nieuczciwy użytkownik. Szum, o odpowiednio dużej mocy, może zniekształcić lub nawet całkowicie zagłuszyć fingerprint osadzony w zaszumionym obrazie. Jednakże, szum o dużej mocy jednocześnie znacznie pogarsza jakość perceptualną obrazu, co powoduje, że ataki te nie są atrakcyjne dla piratów, którzy chcą zachować walory wizualne treści multimedialnej. Aby zwiększyć odporność metody fingerprintingu na ten atak, energia osadzonego fingerprintu powinna być możliwie jak największa. Można to osiągnąć poprzez osadzanie długich fingerprintów lub zwiększanie siły osadzania fingerprintów.

Innym potencjalnym zagrożeniem dla osadzonych fingerprintów może być przetwarzanie sygnału obrazu. Przykładowymi operacjami, które mogą usunąć fingerprint są [20]: konwersja analogowo-cyfrowa oraz cyfrowo-analogowa, ponowne próbkowanie, ponowna kwantyzacja, zmiana kontrastu lub koloru. Są to operacje, które podobnie jak dodawanie szumu powodują pogorszenie jakości perceptualnej multimediów. Natomiast, kompresja stratna lub filtrowanie dolnoprzepustowe są operacjami, które mogą służyć jako skuteczny atak, gdyż powodują utratę wysokoczęstotliwościowych elementów widmowych obrazu, w których mógł być osadzony fingerprint, a jednocześnie nie pogarszają w znacznym stopniu jakości wizualnej. Kompresja jest dla pirata szczególnie atrakcyjnym narzędziem, gdyż możliwe jest łatwe sterowanie poziomem jakości skompresowanego obrazu. Ponadto, kompresja jest często nieunikniona przy przesyłaniu danych multimedialnych, nawet, jeżeli nie zachodzi żaden atak. Z tych względów, wpływ kompresji na osadzone fingerprinty jest często przedmiotem badań w literaturze. Nowe metody JFD dla obrazów kolorowych, przedstawione w rozprawie, również zostały zbadane pod kątem odporności na kompresję. Algorytmem wybranym do badań jest, najczęściej stosowana, kompresja JPEG (ang. *Joint Photographic Experts Group*).

W przypadku obrazów, pirat może wykorzystać przekształcenia geometryczne do uszkodzenia fingerprintu. Jako przekształcenia geometryczne należy rozumieć: obracanie, przesuwanie, kadrowanie, lub skalowanie obrazu. Obracanie i przesuwanie zmieniają kolejność poszczególnych pikseli w liniach obrazu. Wprowadzenie różnic w liniach obrazu względem obrazu oryginalnego powoduje pojawienie się różnic między wydobytym fingerprintem, a fingerprintem pirata, co może uniemożliwić jego identyfikację. Najprostszą ochroną przed tego rodzaju atakami, nazywanych atakami desynchronizacyjnymi, jest zastosowanie detekcji koherentnej, gdyż jeżeli podczas detekcji jest dostępna oryginalna wersja obrazu, to możliwe

jest cofnięcie modyfikacji wprowadzonych przez pirata, poprzez wybranie odpowiadających sobie punktów w obrazie oryginalnym oraz obrazie zniekształconym. Kolejnym atakiem jest kadrowanie (ang. *cropping attack*), które może usunąć fingerprint, lub jego fragment, razem z wykadrowanym fragmentem obrazu. Aby zwiększyć odporność metody fingerprintingu na ten atak, należy osadzać fingerprinty w możliwie jak największej liczbie elementów obrazu, rozproszonych na całej jego powierzchni. Skalowanie natomiast, powoduje efekt podobny do filtrowania dolnoprzepustowego, czyli powoduje utratę wysokoczęstotliwościowych elementów obrazu, w których mógł być osadzony fingerprint. Odpowiednie połączenie różnych przekształceń geometrycznych może stanowić bardzo skuteczny atak.

3.2. Ataki zmowy

Największym zagrożeniem dla osadzonych fingerprintów są ataki zmowy [4, 5, 6, 7, 8] (ang. *collusion attacks*), czyli ataki przeprowadzane przez grupę zmówionych piratów. Każdy z piratów będący w zmowie dysponuje kopią tych samych multimediów, oznakowaną jego fingerprintem. Podczas ataku, piraci wspólnie łączą swoje oznakowane kopie za pomocą pewnej funkcji, by w rezultacie otrzymać kopię piracką, w której fingerprinty piratów są uszkodzone w takim stopniu, że identyfikacja piratów jest bardzo trudna, a w niektórych przypadkach nawet niemożliwa. Ataki zmowy dzieli się na liniowe i nieliniowe, ze względu na funkcję łączącą oznakowane kopie piratów, która może być liniowa albo nieliniowa. Możliwe jest, że połączone fingerprinty piratów w pirackiej kopii mogą zostać tak zniekształcone, że upodobnią się do fingerprintu innego, uczciwego użytkownika. Odporność na ataki zmowy jest największym wyzwaniem w projektowaniu nowych metod fingerprintingu. Zazwyczaj rozpatruje się przypadek, w którym atak zmowy jest realizowany przez wielu nieuczciwych użytkowników, z których każdy dysponuje jedną oznakowaną kopią multimediów. Jednakże, atak zmowy może być również zrealizowany przez jednego użytkownika, który posiada wiele kopii tej samej treści multimedialnej, z których każda jest oznakowana innym fingerprintem.

Z punktu widzenia piratów, skuteczny atak zmowy musi spełniać dwa wymagania. Po pierwsze, każdy zmówiony pirat powinien ponosić jednakowe ryzyko identyfikacji jego tożsamości. Aby to osiągnąć, energia fingerprintów poszczególnych piratów powinna być zredukowana przez taki sam współczynnik odpowiadający liczbie piratów. Wówczas, prawdopodobieństwa detekcji fingerprintów poszczególnych piratów są sobie równe przy założeniu, że energia każdego osadzonego fingerprintu jest taka sama. Po drugie, zmawiający się piraci muszą zadbać o to, by modyfikacje niszczące osadzone fingerprinty nie zniszczyły chronionego obrazu, tzn., aby zmiany wprowadzone w obrazie nie przekroczyły progu zauważalnej różnicy JND. Jak już zauważono na początku tego rozdziału, piratom zależy nie tylko na usunięciu fingerprintu, ale również na zachowaniu jakości wizualnej w pirackiej kopii.

Poza podziałem na ataki zmowy liniowe oraz nieliniowe, w literaturze można znaleźć podział na ataki zmowy I typu i II typu [51, 52, 53]. Zmowa I typu odnosi się do sytuacji, w której ten sam ciąg znakujący jest osadzony w różnych danych. Jest to typowy scenariusz watermarkingu, w którym strona dystrybucyjna podpisuje wydawane przez siebie dane, a osadzony ciąg służy identyfikacji nadawcy. Wówczas, zmawiający się mogą podjąć próbę estymacji watermarku z każdej oznakowanej kopii danych, a następnie otrzymać nieoznakowaną kopię danych przez odjęcie estymaty watermarku od oznakowanej kopii. Zmowa II typu

odnosi się do sytuacji, w której różne ciągi znakujące są osadzone w tych samych danych. Jest to typowy scenariusz fingerprintingu, w którym dane są przypisane do poszczególnych odbiorców, a osadzone ciągi służą identyfikacji odbiorców. Wówczas, zmawiający się mogą podjąć próbę utworzenia nieoznakowanej kopii danych stosując odpowiednią kombinację oznakowanych kopii. W tej pracy rozważane są ataki zmowy II typu.

Możliwe jest zdefiniowanie wielu ataków zmowy, ale w literaturze najczęściej wymienia się następujące [5, 53, 8, 22]: atak uśredniający (ang. *averaging attack*), atak wytnij-wklej (ang. *cut-paste attack*), atak minimum (ang. *minimum attack*), atak maksimum (ang. *maximum attack*), atak medianowy (ang. *median attack*), atak min-max (ang. *min-max attack*), zmodyfikowany atak różnicowy (ang. *modified negative attack*), oraz losowy atak różnicowy (ang. *randomized negative attack*). Poniżej opisano każdy z tych ataków.

Atak uśredniający (ang. *averaging attack*) jest liniowym atakiem zmowy, w którym piraci generują piracką kopię zgodnie z poniższą funkcją [4, 5]:

$$x_n^{\text{ave}} = \sum_{c=1}^{C} \frac{1}{C} x_n^{(c)} , \tag{3.1}$$

gdzie x_n^{ave} jest wartością *n*-tego piksela pirackiej kopii biorącej udział w ataku uśredniającym, $x_n^{(c)}$ jest wartością *n*-tego piksela oznakowanej kopii *c*-tego pirata, $n = 1, 2, ...N$, N jest liczbą wszystkich pikseli obrazu, $c = 1, 2, ...C$, C jest liczbą wszystkich piratów. Atak uśredniający jest bardzo prostym atakiem, ale posiada kilka istotnych zalet. Po pierwsze, w tym ataku każdy pirat, uczestniczący w zmowie, podejmuje jednakowe ryzyko jego wykrycia, ponieważ energia fingerprintu każdego pirata jest redukowana przez taki sam współczynnik 1/*C*. Po drugie, atak uśredniający nie pogarsza jakości wizualnej obrazu. Co więcej, przy rosnącej liczbie piratów w zmowie, obraz będący wynikiem ataku zmierza do obrazu oryginalnego. Po trzecie, w pracach [5, 6, 54] wykazano, że wiele ataków nieliniowych może być aproksymowanych za pomocą ataku uśredniającego z addytywnym szumem. Z powyższych względów, atak uśredniający jest zawsze wykorzystywany do badania odporności metody fingerprintingu na ataki zmowy. W rezultacie, odporność na atak uśredniający stała się powszechnie akceptowanym kryterium porównawczym metod fingerprintingu.

Atak wytnij-wklej (ang. *cut-paste attack*) jest liniowym atakiem zmowy, w którym każdy z *C* zmawiających się piratów wycina inną 1/*C* część swojej oznakowanej kopii, a następnie *C* wyciętych fragmentów obrazu jest składanych w piracką kopię [3]. Wycinane fragmenty muszą być rozłączne i łącznie obejmować całą powierzchnię obrazu. Przy takim ataku, fragment pochodzący od jednego pirata zawiera tylko 1/*C* długości jego fingerprintu przy założeniu, że fingerprint jest równomiernie rozproszony w obrazie. Jeżeli to założenie nie jest spełnione, to część piratów dostarczy do pirackiej kopii więcej bitów swojego fingerprintu niż pozostali piraci, a więc ryzyko wykrycia dla tych piratów będzie wyższe. Atak zmowy wytnij-wklej nie zmienia jakości wizualnej obrazu, tj. jakość pirackiej kopii jest taka sama, jak jakość fragmentów wyciętych z oznakowanych kopii.

Do prostych, nieliniowych ataków zmowy można zakwalifikować atak minimum (ang. *minimum attack*), atak maksimum (ang. *maximum attack*), oraz atak medianowy (ang. *median attack*). W tych atakach, piraci generują pirackie kopie zgodnie z poniższymi wzorami [4, 5]:

$$x_n^{\min} = \min(\{x_n^{(c)}\}_{c=1,2,...C}), \tag{3.2}$$

$$x_n^{\max} = \max(\{x_n^{(c)}\}_{c=1,2,...C}), \tag{3.3}$$

$$x_n^{\text{med}} = \text{med}(\{x_n^{(c)}\}_{c=1,2,...C}), \tag{3.4}$$

gdzie $x_n^{\min}$ jest wartością *n*-tego piksela pirackiej kopii w ataku minimum, $x_n^{\max}$ jest wartością *n*-tego piksela pirackiej kopii w ataku maksimum, x_n^{med} jest wartością *n*-tego piksela pirackiej kopii w ataku medianowym, $x_n^{(c)}$ jest wartością *n*-tego piksela oznakowanej kopii *c*-tego pirata, $n = 1, 2, ...N$, *N* jest liczbą wszystkich pikseli obrazu, $c = 1, 2, ...C$, *C* jest liczbą wszystkich piratów, a funkcje min(·), max(·), oraz med(·) zwracają odpowiednio wartość minimalną, wartość maksymalną, oraz medianę ze zbioru. W ogólności, ataki minimum oraz maksimum nie spełniają wymagania na jednakowe ryzyko detekcji dla wszystkich piratów. Ponadto, ataki minimum oraz maksimum znacznie pogarszają jakość wizualną obrazu. Natomiast, atak medianowy daje podobne rezultaty jak atak uśredniania [5]. Kolejne nieliniowe ataki zmowy są kombinacją powyższych, prostych ataków.

Do złożonych nieliniowych ataków zmowy można zakwalifikować atak min-max (ang. *min-max attack*), zmodyfikowany atak różnicowy (ang. *modified negative attack*), oraz losowy atak różnicowy (ang. *randomized negative attack*). W tych atakach piraci generują piracki kopie zgodnie z poniższymi wzorami [4, 5]:

$$x_n^{\text{minmax}} = \frac{x_n^{\min} + x_n^{\max}}{2}, \tag{3.5}$$

$$x_n^{\text{modneg}} = x_n^{\min} + x_n^{\max} - x_n^{\text{med}}, \tag{3.6}$$

$$x_n^{\text{randneg}} = \begin{cases} x_n^{\min} & \text{z prawd. } \mathrm{P_R} \\ x_n^{\max} & \text{z prawd. } (1-\mathrm{P_R}) \end{cases}, \tag{3.6}$$

gdzie x_n^{minmax} jest wartością *n*-tego piksela pirackiej kopii w ataku min-max, x_n^{modneg} jest wartością *n*-tego piksela pirackiej kopii w zmodyfikowanym ataku różnicowym, x_n^{randneg} jest wartością *n*-tego piksela pirackiej kopii w losowym ataku różnicowym, $x_n^{\min}$ jest wartością *n*-tego piksela w ataku minimum, $x_n^{\max}$ jest wartością *n*-tego piksela w ataku maksimum, x_n^{med} jest wartością *n*-tego piksela w ataku medianowym, $n = 1, 2, ...N$, *N* jest liczbą wszystkich pikseli obrazu, a prawdopodobieństwo $\mathrm{P_R}$ zazwyczaj jest równe ½. Te złożone ataki skuteczniej usuwają fingerprint z obrazu i jednocześnie nie pogarszają jakości wizualnej obrazu. Spośród wszystkich przedstawionych ataków, losowy atak różnicowy jest uznawany za najbardziej efektywny w usuwaniu osadzonych fingerprintów. Zostało to udowodnione w pracy [5].

Nowe metody łącznego fingerprintingu i deszyfracji dla obrazów kolorowych przedstawione w niniejszej rozprawie zostały zbadane pod kątem odporności na ataki zmowy. W przeprowadzonych badaniach zasymulowano: atak uśredniający, atak minimum, atak maksimum, atak medianowy, atak min-max, oraz losowy atak różnicowy. Ponadto, zbadano również odporność proponowanych metod na atak uśredniający połączony z kompresją JPEG.

3.3. Identyfikacja piratów w ataku zmowy

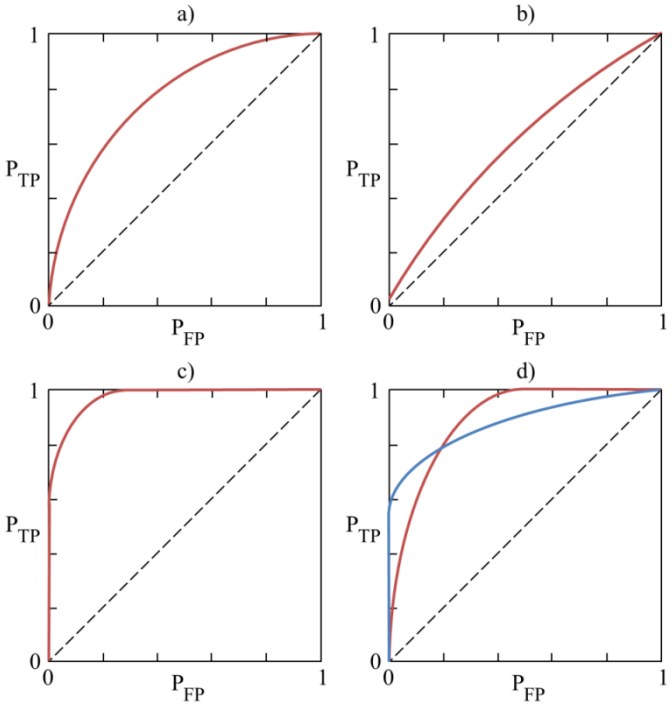
Identyfikację piratów przeprowadza się za pomocą badania korelacji skrośnej fingerprintu wydobytego z pirackiej kopii oraz wszystkich fingerprintów użytkowników. Po przechwyceniu pirackiej kopii i wydobyciu z niej fingerprintu, należy obliczyć współczynniki korelacji między wydobytym fingerprintem i fingerprintami poszczególnych użytkowników. Następnie, obliczone współczynniki korelacji są porównywane z pewną ustaloną wartością, nazywaną progiem detekcji lub progiem identyfikacji. Jeżeli współczynnik korelacji dla fingerprintu *u*-tego użytkownika przekroczy próg identyfikacji, to *u*-ty użytkownik zostaje zidentyfikowany jako pirat. Im niższa jest wartość progu, tym większa jest szansa na zidentyfikowanie piratów, ale również większa jest szansa, że uczciwy użytkownik zostanie uznany za pirata. Zatem, problem poprawnej identyfikacji piratów polega na właściwym określeniu wartości progu identyfikacji. Wybór wartości progu detekcji jest uzależniony od przyjętej strategii wykrywania piratów, która określa pewne kryteria skuteczności identyfikacji piratów. W ogólności, wyróżnia się trzy strategie [3, 4]: „złapać jednego” (ang. *catch one scenario*), „złapać wielu” (ang. *catch meny scenario*), oraz „złapać wszystkich” (ang. *catch all scenario*). Alternatywnie, można analizować funkcję korelacji skrośnej zamiast porównywania wartości współczynników korelacji skrośnej, ale jest to bardziej złożone i niepraktyczne.

Celem strategii „złapać jednego” jest bezbłędna identyfikacja przynajmniej jednego zmawiającego się pirata. W tym scenariuszu, próg identyfikacji jest wybierany w taki sposób, aby maksymalizować szansę zidentyfikowania przynajmniej jednego pirata, bez ryzyka niesłusznego oskarżenia któregokolwiek uczciwego użytkownika. Identyfikacja piratów kończy się niepowodzeniem, gdy detektor nie zidentyfikuje żadnego ze zmawiającym się piratów lub detektor wskaże niewinnego użytkownika jako pirata. Kryteria skuteczności stanowią: minimalne prawdopodobieństwo fałszywie pozytywnej (ang. *false positive*) decyzji oraz minimalne prawdopodobieństwo fałszywie negatywnej (ang. *false negative*) decyzji. Taka strategia wykrywania piratów jest stosowana, kiedy niesłuszne oskarżenia mogą prowadzić do poważnych konsekwencji, np. kiedy gromadzone są materiały dowodowe do rozprawy w sądzie.

Celem strategii „złapać wielu” jest identyfikacja jak największej liczby zmawiających się piratów. W tym scenariuszu, próg identyfikacji jest wybierany w taki sposób, że dopuszczalne jest oskarżenie niewielkiej liczby uczciwych użytkowników, ale oczekuje się, że większość zmawiających się piratów zostanie złapana. Zatem, kryteria skuteczności obejmują oczekiwany odsetek poprawnie zidentyfikowanych piratów w zmowie oraz oczekiwany odsetek niesłusznie oskarżonych użytkowników. Alternatywny zestaw kryteriów obejmuje: akceptowalne prawdopodobieństwo fałszywie pozytywnej (ang. *false positive*) decyzji oraz maksymalne prawdopodobieństwa poprawnie pozytywnej (ang. *true positive*) decyzji. Taka strategia identyfikacji piratów jest stosowana, kiedy wyniki detekcji są połączone z innymi działaniami systemu podejmowania decyzji, np. dalszą obserwacją.

Celem strategii „złapać wszystkich” jest identyfikacja wszystkich zmawiających się piratów. W tym scenariuszu, próg identyfikacji jest wybierany w taki sposób, aby maksymalizować szansę złapania wszystkich piratów i jednocześnie, aby liczba niesłusznie oskarżonych użytkowników nie przekroczyła ustalonego poziomu. Dlatego też, kryteria skuteczności dla tego scenariusza obejmują: maksymalne prawdopodobieństwo wykrycia wszystkich piratów oraz akceptowalny wskaźnik efektywności, utożsamiany z oczekiwaną liczbą niesłusznie oskarżonych użytkowników, przypadającą na każdego pirata w zmowie. Ta strategia wykrywania piratów jest używana w przypadku, gdy wiarygodność odbiorców informacji jest kwestią wielkiej wagi oraz istnieje konieczność identyfikacji wszystkich piratów za wszelką cenę.

Po wybraniu strategii wykrywania piratów, należy wybrać wartość progu identyfikacji w taki sposób, aby przyjęte kryteria skuteczności zostały spełnione. W tym celu można posłużyć się przeprowadzonymi badaniami i funkcją charakterystyki operacyjnej detektora ROC (ang. *Receiver Operating Characteristics*) [55]. Funkcja ROC jest narzędziem opisującym skuteczność detektora, wykorzystywanym w badaniach skuteczności podejmowania decyzji m.in. przez detektory steganograficzne, algorytmy sztucznej inteligencji, oraz detektory cyfrowych znaków wodnych. Charakterystyka ROC jest zdefiniowana jako zależność prawdopodobieństwa poprawnie pozytywnego (ang. *true positive*) wyniku P_{TP} w funkcji prawdopodobieństwa fałszywie pozytywnego (ang. *false positive*) wyniku P_{FP}, czyli funkcja $P_{TP}(P_{FP})$. Przykładowe krzywe charakterystyki ROC zostały przedstawione na rysunku 3.2. Krzywa ROC nieskutecznego detektora jest zbliżona do przekątnej wykresu (Rys. 3.2. (b)), gdyż prosta diagonalna $P_{TP} = P_{FP}$ oznacza charakterystykę całkowicie losowego podejmowania decyzji. Natomiast, krzywa ROC skutecznego detektora jest zbliżona do górnego lewego rogu wykresu (Rys. 3.2. (c)), gdyż prosta $P_{TP} = 1$ oznacza charakterystykę bezbłędnego detektora.



Rys. 3.2. a) Przykładowa charakterystyka ROC, b) charakterystyka ROC nieskutecznego detektora, c) charakterystyka ROC skutecznego detektora, d) charakterystyki ROC detektorów trudnych do porównania, opracowanie własne na podstawie [55].

Porównanie skuteczności dwóch detektorów polega na porównaniu ich funkcji ROC. Jeżeli $P_{TP}^{(A)}(P_{FP}^{(A)})$ oznacza funkcję ROC detektora A, a $P_{TP}^{(B)}(P_{FP}^{(B)})$ oznacza funkcję ROC detektora B oraz $P_{TP}^{(A)}(P_{FP}^{(A)}) > P_{TP}^{(B)}(P_{FP}^{(B)})$ dla $P_{FP} \in <0,1>$, to detektor A jest bardziej skuteczny niż detektor B. Niestety, w ogólności funkcje ROC różnych detektorów mogą się przecinać (Rys. 3.2. (d)) i ich porównanie nie jest tak oczywiste. W takich przypadkach, detektory mogą zostać porównane poprzez: porównanie powierzchni pod krzywymi ROC poszczególnych detektorów, porównanie powierzchni między krzywymi ROC detektorów, a prostą diagonalną, lub porównanie wartości funkcji ROC (prawd. P_{TP}) poszczególnych detektorów dla zadanego argumentu (prawd. P_{FP}).

Dalej w rozprawie, w celu analizy odporności na ataki zmowy proponowanych nowych metod JFD wybrano strategię „złapać wielu", a jako kryterium oceny skuteczności identyfikacji piratów przyjęto akceptowalne prawdopodobieństwo fałszywie pozytywnej (ang. *false positive*) decyzji P_{FP} oraz maksymalne prawdopodobieństwo poprawnie pozytywnej (ang. *true positive*) decyzji P_{TP}. Przyjęto, że poprawnie pozytywna decyzja oznacza, że pirat uczestniczący w zmowie został zidentyfikowany, natomiast fałszywie pozytywna decyzja oznacza, że uczciwy użytkownik został niesłusznie oskarżony. Próg identyfikacji został wybrany w taki sposób, aby osiągnąć akceptowalne prawdopodobieństwo oskarżenia niewinnego użytkownika P_{FP}. W celu porównania skuteczności wykrywania piratów zaprojektowanych, nowych metod JFD obliczono ich charakterystyki ROC. Przy ustalonym prawdopodobieństwie P_{FP} i prawdopodobieństwie P_{TP} odczytanym z charakterystyki ROC, określono maksymalną liczbę piratów w zmowie, na którą są odporne poszczególne nowe metody JFD.

3.4. Osadzanie fingerprintów w dziedzinie częstotliwości

W celu osiągnięcia wysokiej odporności systemu fingerprintingu na ataki, konieczne jest wybranie odpowiedniej strategii konstrukcji i osadzania fingerprintów. Kryteria konstrukcji zbioru fingerprintów określono już w rozdziale 2.2., natomiast w tym podrozdziale wyjaśniono, w których elementach obrazu należy osadzać fingerprinty. Najprostszym sposobem jest osadzanie fingerprintów w wartościach pikseli obrazu, czyli w tzw. dziedzinie przestrzennej (ang. *spatial domain*) [56, 57, 13]. Jednakże, prostota takiego rozwiązania jest jego jedyną zaletą, gdyż fingerprinty osadzone w dziedzinie przestrzennej są względnie łatwe do usunięcia oraz najczęściej występują tylko we fragmencie chronionego obrazu. W tym podrozdziale, przedstawiono inne podejście, czyli osadzanie fingerprintów we współczynnikach wybranej transformaty, czyli w dziedzinie częstotliwości (ang. *frequency domain*). Taka koncepcja jest również nazywana osadzaniem z rozpraszaniem widma (ang. *spread spectrum watermarking/fingerprinting*) i po raz pierwszy została przedstawiona przez Coxa i innych [20]. W tym podrozdziale, zagadnienie osadzania fingerprintów w dziedzinie częstotliwości zostało przedstawione w kontekście obrazów.

Z punktu widzenia ochrony przed atakami, istotne jest, aby każda próba usunięcia lub zniszczenia fingerprintu powodowała znaczącą degradację jakości wizualnej chronionych danych jeszcze zanim fingerprint zostanie zniszczony. Wówczas, gdy pirat usunie fingerprint z danych, to chronione dane utracą swoją wartość. W szczególności, fingerprint powinien być

odporny na standardowe operacje przetwarzania sygnałów, geometryczne przekształcenia obrazu i ataki zmowy. W celu osiągnięcia pożądanej, wysokiej odporności na usunięcie, zaleca się osadzanie fingerprintów w perceptualnie najbardziej istotnych składowych obrazu (ang. *perceptually most significant components*) [20]. W takim przypadku, w celu usunięcia fingerprintu, pirat musi zaatakować właśnie te fundamentalne składowe obrazu, co nieuchronnie prowadzi do pogorszenia jego jakości wizualnej. Z początku, takie podejście może wydawać się sprzeczne z intuicją, ponieważ można twierdzić, że skoro modyfikacja perceptualnie istotnych elementów prowadzi do uszkodzenia obrazu, to nie jest możliwe osadzenie w nich fingerprintów w sposób niezauważalny dla ludzkiego oka. Jednakże, omawiane istotne elementy posiadają pewną perceptualną pojemność (ang. *perceptual capacity*), która pozwala na osadzenie fingerprintu bez pogorszenia jakości obrazu [20].

W kontekście osadzania fingerprintów w dziedzinie częstotliwości, jako istotne elementy należy rozumieć istotne widmowe elementy sygnału obrazu, a konkretniej niskoczęstotliwościowe elementy widmowe. Większość technik przetwarzania sygnału, które mogą być również użyte jako ataki, pozostawiają elementy istotne perceptualnie w stanie stosunkowo nienaruszonym. Na przykład, kompresja stratna zazwyczaj usuwa tylko perceptualnie nieistotne elementy obrazu. Realizowane jest to przez usunięcie wysokoczęstotliwościowych składowych widmowych z wykorzystaniem wybranej transformaty kompresowanego sygnału. Podobnie, skalowanie (zmniejszanie) obrazu powoduje utratę wysokoczęstotliwościowych elementów widmowych. Kadrowanie lub wycinanie fragmentów obrazu może poważnie uszkodzić lub nawet całkowicie usunąć fingerprint osadzony przestrzennie. Natomiast fingerprinty osadzone w dziedzinie częstotliwości, po powrocie do dziedziny przestrzennej, są równomiernie rozproszone na całej powierzchni obrazu, zatem wpływ kadrowania (ang. *cropping attack*) jest znacznie mniejszy.

Powyższa dyskusja wskazuje, że fingerprint nie może być osadzony w elementach nieistotnych perceptualnie, gdyż każda operacja powodująca bezpośrednio lub pośrednio filtrowanie dolnopasmowe może usunąć tak osadzony fingerprint przy stosunkowo niewielkiej degradacji jakości obrazu. Problem polega na tym, że konieczne jest znalezienie sposobu na osadzenie fingerprintu w elementach istotnych perceptualnie z zachowaniem jakości wizualnej chronionych danych. Innymi słowy, konieczne jest określenie pojemności perceptualnej tych komponentów. Oczywiście, każdy element widmowy może być zmodyfikowany w sposób niezauważalny, jeżeli energia modyfikacji (osadzonego fingerprintu) jest niewielka, aczkolwiek małe zmiany są bardzo wrażliwe na szum addytywny. Możliwe jest zwiększenie energii osadzonego fingerprintu w pewnych elementach widmowych, przy wykorzystaniu wiedzy na temat zjawiska maskowania w ludzkich narządach wzroku i słuchu, które są najbardziej wrażliwe na niskoczęstotliwościowe składowe widmowe o dużej energii [58].

W celu modyfikacji częstotliwościowych elementów obrazu, wykorzystuje się wybraną metodę transformacji obrazu. W ogólności, fingerprinty mogą być osadzane we współczynnikach dowolnej transformaty, np. dyskretnej transformaty kosinusowej DCT (ang. *Discrete Cosine Transform*) [9, 59, 60, 61] lub dyskretnej transformaty falkowej DWT (ang. *Discrete Wavelet Transform*) [62, 63, 64]. Przy zastosowaniu odpowiedniej transformacji, obliczana jest tzw. maska perceptualna (ang. *perceptual mask*) oraz siła osadzania fingerprintu określające, w których elementach transformaty możliwe jest osadzenie fingerprintu bez pogorszenia jakości perceptualnej chronionych danych. Zmiany wprowadzone w obrazie powinny być

mniejsze od progu zauważalnej różnicy JND (ang. *Just Noticeable Difference*) [65]. Dokładna amplituda modyfikacji wprowadzonych w tych elementach jest znana jedynie stronie dystrybucyjnej. Atakujący pirat może posiadać jedynie wiedzę na temat możliwego zakresu zmian. Aby atakujący miał pewność, że fingerprint został usunięty, musi założyć, że amplituda każdej wprowadzonej modyfikacji była równa największej wartości z możliwego zakresu, pomimo, że zazwyczaj liczba tak dużych modyfikacji jest niewielka. W rezultacie, atak powoduje widoczne defekty w obrazie, co nie jest pożądane z punktu widzenia pirata. Z tego względu, można twierdzić, że osadzony fingerprint jest odporny na usunięcie.

Nowe metody JFD przedstawione w niniejszej rozprawie osadzają fingerprinty w dziedzinie częstotliwości za pomocą transformacji DCT [17, 18]. Zakłada się, że każdy współczynnik transformaty posiada pewną pojemność perceptualną, tzn. pewna ilość informacji może być dodana bez wpływu (lub z minimalnym wpływem) na jakość wizualną obrazu. W proponowanych metodach JFD, w celu osadzenia fingerprintu o długości *M*, obraz zostaje podzielony na bloki 8 × 8 pikseli i każdy blok jest poddawany transformacji DCT. Następnie, fingerprint jest osadzany w *M* współczynnikach DCT o największej energii, z wyłączeniem współczynników DC (ang. *zero-frequency component*). Zazwyczaj, są to współczynniki niskiego rzędu, skupione w górnym lewym rogu bloku, które odpowiadają niskoczęstotliwościowym elementom widmowym, czyli perceptualnie istotnym elementom obrazu [20, 66].

3.5. Podsumowanie

W tym rozdziale zostały opisane zagrożenia dla osadzonych fingerprintów. Zagrożenia te podzielono na ataki realizowane przez jednego pirata oraz ataki zmowy realizowane przez grupę piratów. Wśród zagrożeń od strony pojedynczego pirata wymieniono: dodawanie szumu do oznakowanej kopii, mechanizmy przetwarzania sygnałów, oraz przekształcenia geometryczne oznakowanego obrazu. Natomiast, w przypadku zagrożeń od strony grupy piratów opisano liczne ataki zmowy, zarówno liniowe jak i nieliniowe. Wyjaśniono również różnicę między atakiem zmowy typu I i typu II.

Do badania odporności projektowanych metod fingerprintingu, wybrano następujące ataki: atak za pomocą szumu, atak za pomocą kompresji, ataki zmowy (atak uśredniający, atak maksimum, atak minimum, atak medianowy, atak min-max, losowy atak różnicowy) oraz uśredniający atak zmowy połączony z kompresją. W ostatnim podrozdziale, opisano możliwe strategie detekcji piratów, a także zdefiniowano funkcję charakterystyki operacyjnej detektora ROC. Dla badań jakości proponowanych nowych metod JFD wybrano strategię „złapać wielu", a jako kryteria oceny skuteczności identyfikacji piratów przyjęto prawdopodobieństwo fałszywie pozytywnej (ang. *false positive*) decyzji P_{FP} oraz prawdopodobieństwo poprawnie pozytywnej (ang. *true positive*) decyzji P_{TP}.

W końcowej części rozdziału zostały omówione zalety osadzania fingerprintów we współczynnikach transformaty, czyli w dziedzinie częstotliwości oraz podkreślono przewagę takiego postępowania nad osadzaniem fingerprintów w pikselach obrazu, czyli w dziedzinie przestrzeni. Transformatą wykorzystywaną w nowych metodach JFD zaproponowanych w tej rozprawie jest dyskretna transformata kosinusowa DCT, wybrana z tego względu, że jest ona stosowana w popularnych standardach kompresji JPEG oraz MPEG.

4. ŁĄCZNY FINGERPRINTING I DESZYFRACJA

W masowej dystrybucji treści multimedialnych, niezwykle istotne jest osiągnięcie efektywnej wysokiej efektywności, która zminimalizuje koszty utrzymania systemu. Jako rozwiązania efektywne należy zrozumieć rozwiązania o dużej wydajności konstrukcji oraz wydajności dystrybucji. Pojęcia te zostały wyjaśnione w rozdziale 2. Zatem, poszukiwane są rozwiązania, w których złożoność obliczeniowa oraz wymagania na dostępną przepustowość korzystnie skalują się wraz z rosnącą liczbą użytkowników. W celu osiągnięcia pożądanych właściwości wydajnościowych, projektuje się wiele metod fingerprintingu wykorzystujących połączenia typu multicast. W rozdziale 2. opisano różne podejścia do dystrybucji oznakowanych multimediów za pomocą transmisji multicast, natomiast ten rozdział poświęcono koncepcji łącznego fingerprintingu i deszyfracji JFD (ang. *Joint Fingerprinting and Decryption*), która pozwala na największą oszczędność zasobów obliczeniowych i sieciowych.

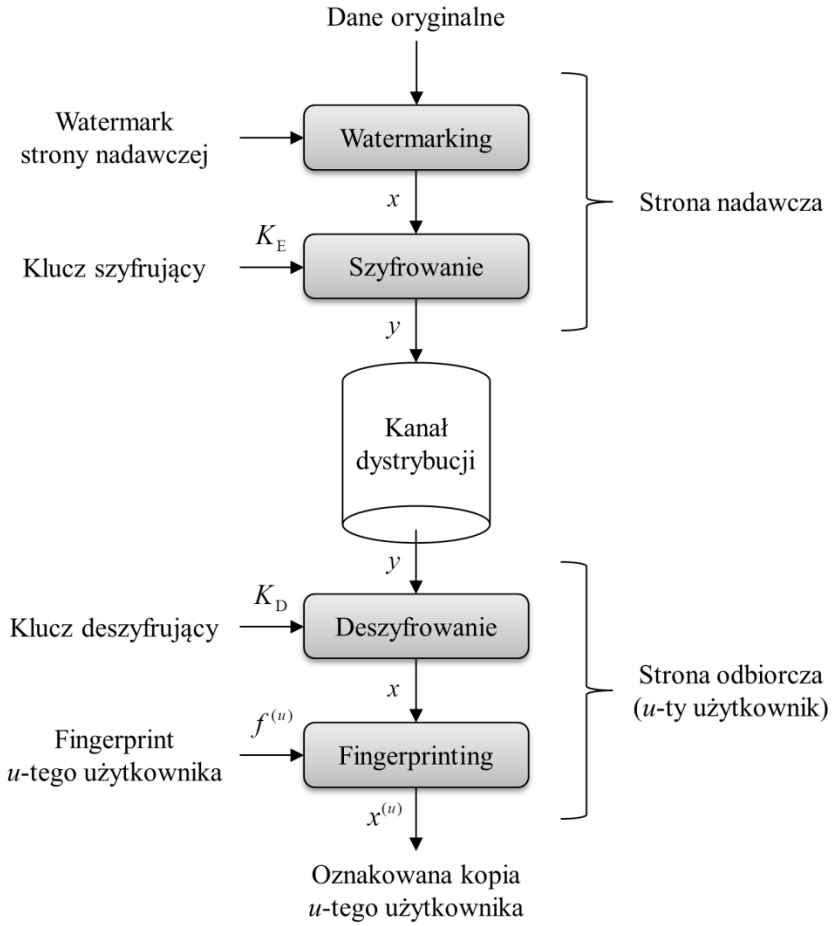
Metody łącznego fingerprintu i deszyfracji są metodami, które łączą w sobie aspekty fingerprintingu i szyfrowania. Jednakże należy pamiętać, że głównym celem metod JFD jest identyfikacja nieuczciwych użytkowników, którzy nielegalnie rozpowszechniają dane multimedialne, a nie zapewnienie poufności przesyłanych danych. W tym rozdziale, najpierw zostanie opisana geneza oraz zasady działania metod JFD. Następnie, przedstawiono kwestie bezpieczeństwa kryptograficznego w kontekście JFD. Na końcu zamieszczono przegląd metod JFD związanych z rozprawą, które miały wpływ na powstanie nowych metod zaproponowanych w rozprawie i opisanych w kolejnych rozdziałach.

4.1. Zasady działania metod JFD

Najbardziej charakterystyczną cechą metod łącznego fingerprintingu i deszyfracji jest to, iż strona dystrybucyjna rozsyła jednakową wersję danych multimedialnych do wszystkich użytkowników, a fingerprinty są osadzane w danych po stronie odbiorczej. Ponieważ identyczne dane są wysyłane do wszystkich użytkowników, to możliwe jest wykorzystanie jednej transmisji multicast w celu dystrybucji multimediów. Dzięki temu, zmniejszone zostają wymagania na dostępną przepustowość, co prowadzi do wysokiej wydajności dystrybucji. Ponadto, w metodach JFD strona dystrybucyjna szyfruje dane tylko jeden raz, w odróżnieniu do klasycznego modelu, gdzie w każdej kopii osadzany jest fingerprint i następnie jest ona szyfrowana, niezależnie dla każdego odbiorcy. Dzięki temu zmniejszone zostają wymagania na moc obliczeniową strony nadawczej, co prowadzi do wysokiej wydajności konstrukcji.

Geneza takiej dystrybucji multimediów i koncepcji osadzania fingerprintu po stronie odbiorcy jest związana z systemami DRM stosowanymi w kinie cyfrowym. Schemat działania takiego systemu został przedstawiony na rysunku 4.1. Na początku, autor multimediów dostarcza oryginalne dane do dystrybutora, który zazwyczaj osadza w nich swój znak wodny (watermark) identyfikujący źródło danych. Następnie, dane multimedialne *x* są szyfrowane za pomocą klucza szyfrującego K_{E}. Szyfrogram *y* jest rozsyłany multicastowo do wszystkich zarejestrowanych użytkowników. Po przejściu przez kanał dystrybucji, następuje deszyfracja danych *y* z użyciem klucza deszyfrującego K_{D}. W rezultacie otrzymywane są oryginalne dane multimedialne *x* z opcjonalnym watermarkiem strony nadawczej, jeżeli taki był osadzony. W takim pierwotnym rozwiązaniu, zazwyczaj każdy odbiorcza dysponuje identycznym klu-

czem. Następnie, w danych x zostaje osadzony fingerprint u-tego odbiorcy $f^{(u)}$. Dopiero po osadzeniu fingerprintu, odbiorca posiada swoją unikatowo oznakowaną kopię $x^{(u)}$, której wolno mu używać.

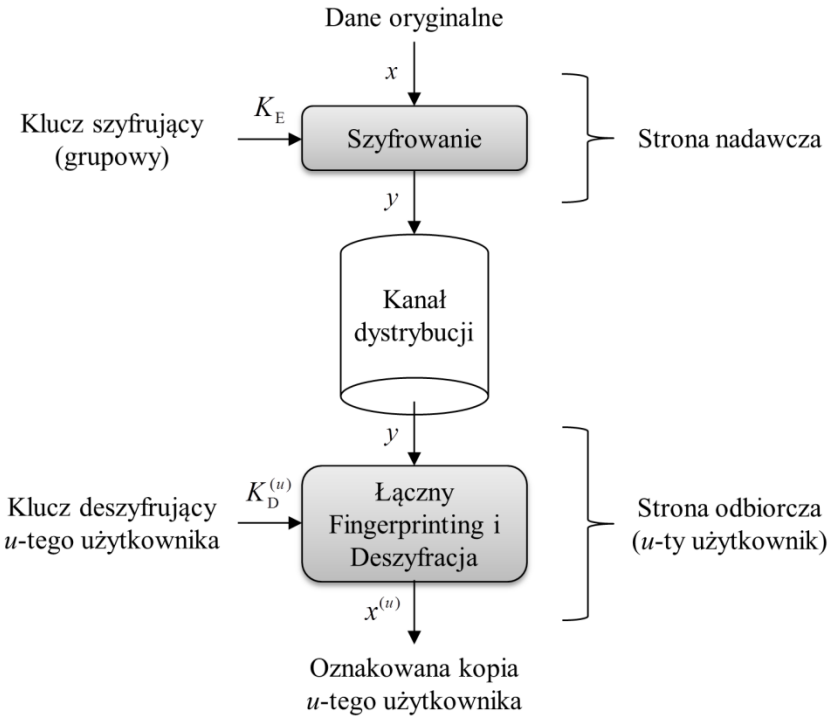


Rys. 4.1. Schemat osadzania fingerprintu po stronie odbiorcy, opracowanie własne.

Z powyższego rysunku i opisu wynika, że odbiorca ma dostęp do nieoznakowanych danych oryginalnych x. W celu ograniczenia dostępu do danych bez fingerprintu, deszyfrowanie i fingerprinting są zazwyczaj zintegrowane w układzie scalonym, który jest zabezpieczony przed ingerencją użytkowników. Takie rozwiązanie jest problematyczne i kosztowne, a ponadto bezpieczeństwo takiego systemu opiera się na zaufaniu do strony odbiorczej. Podejście takie jest możliwe w kinie cyfrowym, gdzie można umieścić specjalistyczne urządzenia i gdzie pracuje zaufany personel. Takie środowisko można uznać za względnie bezpieczne. Jednakże, to podejście nie jest dopuszczalne w usługach VoD o zasięgu globalnym, realizowanych przez Internet. Stosowanie specjalnych urządzeń dekodujących STB (ang. *Set-Top Box*) po stronie odbiorczej zwiększa koszt systemu oraz utrudnia użytkownikom dostęp do treści multimedialnych, co potencjalnie prowadzi do zmniejszenia liczby chętnych do otrzymania takiej usługi.

Łączny fingerprinting i deszyfracja to udoskonalenie modelu stosowanego w kinie cyfrowym. Deszyfracja i fingerprinting zostają zintegrowane w jeden nierozłączny proces, w taki sposób, że nie jest możliwe uzyskanie dostępu do danych nieoznakowanych x po stro-

nie odbiorczej. W rezultacie, urządzenia dekodujące STB oraz zaufanie do użytkowników nie są konieczne, dzięki czemu możliwe jest wykorzystanie takiego rozwiązania w sieci Internet. Ponadto, liczba obliczeń wykonywanych po stronie odbiorczej jest zmniejszona, ponieważ fingerprint jest osadzany w trakcie procesu deszyfrowania. Jednocześnie, integracja procesu fingerprintingu i deszyfracji nie powoduje wzrostu zapotrzebowania na przepustowość kanału dystrybucji, ani wzrostu liczby operacji wykonywanych przez stronę nadawczą.



Rys. 4.2. Ogólna zasada działania łącznego fingerprintingu i deszyfracji, opracowanie własne.

Ogólna zasada działania łącznego fingerprintingu i deszyfracji została zilustrowana na rysunku 4.2. Najpierw, autor multimediów dostarcza dane oryginalne *x* stronie dystrybucyjnej. Oczywiście, dystrybutor może osadzić w nich swój watermark, jednakże nie jest to niezbędna część metody JFD. Następnie, dane multimedialne *x* są szyfrowane za pomocą klucza szyfrującego K_E. Klucz ten jest również nazywany kluczem grupowym, ponieważ wykorzystywany jest jeden klucz K_E do obsługi całej grupy odbiorców. Szyfrowanie jest jedyną operacją wykonywaną przez stronę nadawczą. Szyfrogram *y* jest rozsyłany, za pomocą transmisji multicast, do wszystkich zarejestrowanych użytkowników. Po przejściu przez kanał dystrybucji, następuje deszyfracja danych *y* z użyciem klucza deszyfrującego $K_D^{(u)}$, który jest unikatowym kluczem *u*-tego użytkownika i różni się od klucza szyfrującego K_E. Różnice między kluczami K_E oraz $K_D^{(u)}$ powodują, że w procesie deszyfracji wprowadzane są pewne zmiany w obrazie zdeszyfrowanym w stosunku do obrazu oryginalnego. Zmiany te są niezauważalne dla ludzkiego oka oraz są unikatowe w skali wszystkich użytkowników, zatem są fingerprintami. W rezultacie otrzymywane są oznakowane dane multimedialne $x^{(u)}$ z osadzonych fingerprintem $f^{(u)}$ *u*-tego użytkownika. Użytkownik nie ma dostępu do danych oryginalnych *x*, ponieważ fingerprint jest osadzany w trakcie deszyfrowania.

Należy zwrócić uwagę, iż pomimo stosowania różnych kluczy do szyfrowania i deszyfrowania danych, algorytm szyfrowania stosowany w metodach JFD nie należy do kryptografii asymetrycznej. Różnice między kluczem szyfrującym K_{E} i deszyfrującym $K_D^{(u)}$ wynikają z integracji procesów fingerprintingu i deszyfracji, a wykorzystywany algorytm szyfrowania zazwyczaj należy do kryptografii symetrycznej. W związku z tym, klucze deszyfrujące muszą być dostarczone w bezpieczny sposób do poszczególnych użytkowników.

4.2. Bezpieczeństwo szyfrowania w JFD

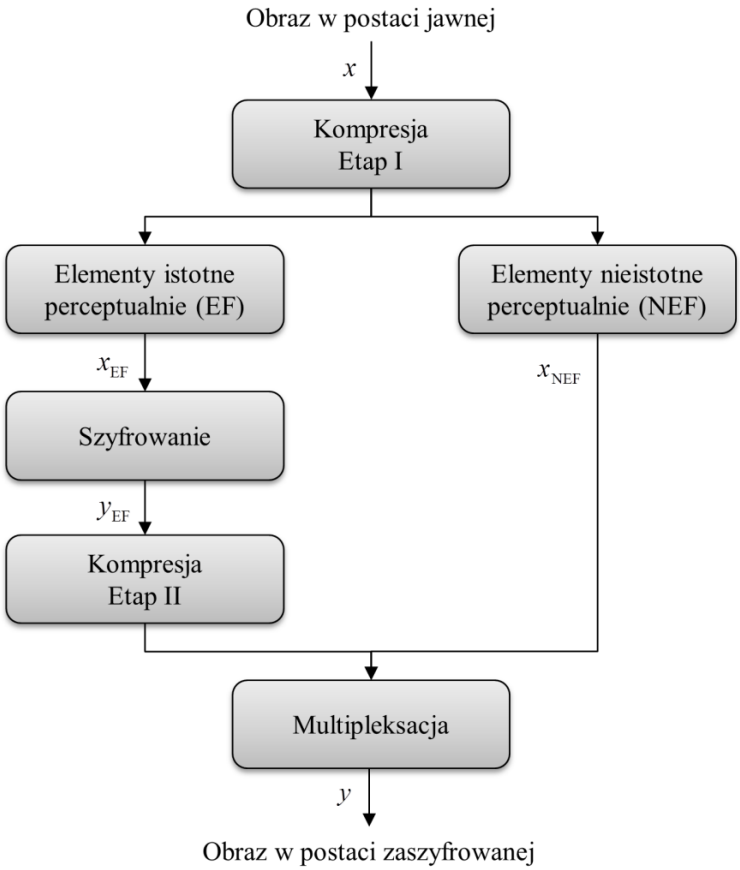
Pojęcie bezpieczeństwa całego systemu fingerprintingu można podzielić na trzy zagadnienia [67]: bezpieczeństwo multimediów, bezpieczeństwo kryptograficzne, oraz bezpieczeństwo fingerprintingu. Bezpieczeństwo w kontekście multimediów oznacza, iż system powinien ograniczać nieupoważnionym użytkownikom dostęp do dystrybuowanych multimediów. Bezpieczeństwo w kontekście kryptograficznym oznacza, iż nieupoważnieni użytkownicy nie mogą posiadać informacji pozwalającej na złamanie algorytmu szyfrowania. Natomiast, bezpieczeństwo w kontekście fingerprintingu oznacza, iż użytkownicy systemu nie mogą mieć dostępu do nieznakowanej kopii multimediów, tzn. kopii bez fingerprintu. Jeżeli powyższe wymagania będą spełnione, to system jest uważany za bezpieczny.

Należy zwrócić uwagę, że bezpieczeństwo szyfrowania multimediów, zwłaszcza obrazów, różni się od bezpieczeństwa systemu kryptograficznego w klasycznym rozumieniu. Rozróżnia się niski i wysoki poziom bezpieczeństwa szyfrowania obrazów [67]. Przy niskim poziomie bezpieczeństwa (ang. *low level security*), zaszyfrowany obraz charakteryzuje się znacznie pogorszoną jakością perceptualną obrazu w porównaniu do obrazu oryginalnego. Zawartość obrazu jest nadal częściowo widoczna po zaszyfrowaniu, ale dane nie mają wartości, gdyż pogorszenie jakości jest tak znaczące, że dane nie nadają się do używania zgodnie z pierwotnym przeznaczeniem. W pracy [68] takie zabezpieczenie nazwano niepełną poufnością (ang. *incomplete confidentiality*). Przy wysokim poziomie bezpieczeństwa (ang. *high level security*), zaszyfrowany obraz wygląda jak szum. Zawartość obrazu jest zupełnie niewidoczna po zaszyfrowaniu i całkowicie niezrozumiała dla oglądających. Jednakże, pełne szyfrowanie obrazu trwa znacznie dłużej niż częściowe szyfrowanie, a szybkość szyfrowania jest krytycznym parametrem. Z punktu widzenia bezpieczeństwa multimediów, wystarczające jest wykorzystanie niskiego poziomu bezpieczeństwa. Zazwyczaj, obraz jest szyfrowany poprzez zasłanianie (ang. *obscuring*) najistotniejszych elementów obrazu, w taki sposób, aby jakość zaszyfrowanego obrazu nie pozwalała nieupoważnionemu użytkownikowi na komfortowe oglądanie chronionych treści multimedialnych.

W klasycznym modelu fingerprintingu, przedstawionym w rozdziale 2., szyfrowanie nie jest częścią fingerprintingu i w ogólności może być wybrane dowolnie. Natomiast, w metodach JFD szyfrowanie jest częścią procesu fingerprintingu i zazwyczaj są to algorytmy słabsze w porównaniu do np. takich szyfrów jak DES, czy AES. Należy zwrócić uwagę, iż jest to celowy i świadomy wybór, gdyż tak silne szyfry posiadają właściwości, które uniemożliwiają realizację łącznego fingerprintingu i deszyfracji. Taką właściwością jest charakterystyczny dla silnych szyfrów tzw. efekt lawinowy (ang. *avalanche effect*), który powoduje, że w wyniku zmiany dowolnego bitu w kluczu, każdy bit obrazu odszyfrowanego może zmienić swój

stan z prawdopodobieństwem równym ½. Jest to sprzeczne z koncepcją JFD, w której różnice między kluczami mają wprowadzić niezauważalne zmiany (fingerprinty) w obrazie odszyfrowanym, a nie całkowicie uniemożliwić jego deszyfrację.

Szyfrowanie wideo wymaga szczególnego podejścia, gdyż oznacza szyfrowanie bardzo dużych ilości informacji. W połączeniu z wymaganiami na masową dystrybucją usług VoD oraz transmisję w czasie rzeczywistym w usługach strumieniowania wideo, szyfrowanie wideo napotyka poważne ograniczenia ze strony ograniczonych zasobów obliczeniowych zarówno strony nadawczej, jak i odbiorczej. W pracach [9, 69] przedstawiono podejście częściowego szyfrowania (ang. *partial encryption*) dla obrazów i wideo, którego celem jest zmniejszenie liczby obliczeń oraz wielkości opóźnienia związanych z szyfrowaniem. Szyfrowanie częściowe jest również nazywane szyfrowaniem selektywnym. W tym podejściu, szyfrowany jest niewielki podzbiór danych, a proces szyfrowania jest połączony z kompresją.



Rys. 4.3. Koncepcja częściowego szyfrowania wideo, opracowanie własne na podstawie [9].

Koncepcja częściowego szyfrowania została przedstawiona na rysunku 4.3. Najpierw, następuje częściowa kompresja (etap I) obrazu oryginalny x, np. kwantyzacja współczynników transformaty obrazu, a następnie zostaje on podzielony na zbiór elementów istotnych perceptualnie EF (ang. *Essential Features*) oraz zbiór elementów nieistotnych perceptualnie NEF (ang. Non-Essential Features). Elementy istotne x_{EF} są przeznaczone do szyfrowania, a tymczasem elementy nieistotne x_{NEF} pozostaną w swojej jawnej postaci. Następnie, zaszyfrowane elementy istotne y_{EF} zostają skompresowane (etap II) z wykorzystaniem kompresji

bezstratnej. Wynik tej kompresji zostaje połączony z elementami nieistotnymi perceptualnie i w rezultacie otrzymywany jest zaszyfrowany i skompresowany obraz *y*.

Najważniejszym etapem w częściowym szyfrowaniu obrazów i wideo jest podział danych na elementy istotne oraz nieistotne perceptualnie. W rozdziale 3.4. wyjaśniono, że podział ten można skojarzyć z podziałem na wysokoczęstotliwościowe i niskoczęstotliwościowe elementy obrazu. Jednakże, aby uzyskać pożądaną redukcję złożoności obliczeniowej oraz redukcję opóźnienia szyfrowania przy jednoczesnym akceptowalnym ukryciu zawartości obrazu, należy spełnić następujące kryteria [9, 69]. Po pierwsze, w celu ograniczenia złożoności obliczeniowej, zbiór elementów przeznaczonych do szyfrowania powinien być możliwie mało liczny. Oznacza to, że zbiór elementów istotnych x_{EF} powinien być o wiele mniej liczny od zbioru elementów nieistotnych x_{NEF}. Po drugie, ponieważ bezpieczeństwo multimediów związane jest z jakością wizualną obrazu zaszyfrowanego, podział na elementy x_{EF} i x_{NEF} powinien gwarantować, że jakość zaszyfrowanego obrazu jest znacząco i bezwarunkowo gorsza od jakości obrazu oryginalnego. Oznacza to, że x_{EF} powinien zawierać większość krytycznie istotnych elementów obrazu. Po trzecie, estymacja elementów x_{EF} na podstawie znajomości x_{NEF} powinna być niemożliwa. W przeciwnym przypadku, pirat mający dostęp do estymaty zbioru x_{EF} oraz do zaszyfrowanej postaci y_{EF} mógłby przeprowadzić atak ze znanym tekstem otwartym. Po czwarte, jeżeli kompresja jest oparta o pewien standard kodowania, to elementy x_{EF} i x_{NEF} powinny być łatwo dostępnymi elementami, związanymi z wykorzystywanym algorytmem kompresji.

Powyższe założenia dotyczące poziomu bezpieczeństwa szyfrowania oraz idea szyfrowania selektywnego (ang. *partial encryption*) zostały przyjęte podczas projektowania nowych metod JFD dla obrazów kolorowych. W tych metodach, podobnie jak w [9, 69], podział elementów obrazu został zrealizowany w oparciu o współczynniki transformaty DCT. Należy zwrócić uwagę, że nie jest to jedyny sposób podziału obrazu dla szyfrowania selektywnego. Na przykład, w pracy [70] przedstawiono szyfrowanie selektywne ze względu na wybrane piksele (ang. *pixels of interest*) oraz dekompozycję SVD (ang. *Singular Value Decomposition*).

4.3. Przegląd literatury metod JFD

W ostatnich latach zaproponowano wiele metod łącznego fingerprintingu i deszyfracji. W tym podrozdziale został przedstawiony przegląd najważniejszych prac, przy czym szczególną uwagę poświęcono pracom, które miały wpływ na rozwój nowych metod JFD zaprezentowanych w rozdziałach 5., 7., i 8.

Pierwszą metodą, zgodną z koncepcją JFD, była metoda Chameleon zaproponowana przez Andersona i Manifavasa [10]. Chameleon jest szyfrem strumieniowym, przeznaczonym do osadzania fingerprintów w danych audio. Klucz deszyfrujący użytkownika składa się z dwóch części: pierwsza część klucza jest ściśle związana z zastosowanym szyfrem strumieniowym, natomiast druga część jest tablicą odniesienia (ang. *lookup-table*) zbudowaną z 2^{16} 64-bitowych słów. Tak długie klucze są niewątpliwą wadą tego rozwiązania. Podczas deszyfrowania fingerprint użytkownika jest osadzany w najmniej znaczących bitach danych. Nie-

zauważalność fingerprintu zostaje osiągnięta, gdyż jakość perceptualna oznakowanej kopii nie ulega istotnemu pogorszeniu, jednakże osadzone w ten sposób fingerprinty są bardzo wrażliwe nawet na najprostsze ataki, np. dodanie szumu lub kompresja dźwięku. Należy zwrócić uwagę, że Chameleon operuje na danych w postaci nieskompresowanej. Ponadto, odporność metody Chameleon na ataki zmowy jest bardzo niska. Autorzy podają [10], że metoda jest w stanie zidentyfikować zaledwie do 4 zmawiających się piratów. Pomimo powyższych wad, metoda Chameleon była inspiracją dla wielu autorów późniejszych rozwiązań.

Kundur i Karthnik po raz pierwszy przedstawili koncepcję łącznego fingerprintingu i deszyfracji w pracy [9] opisującej metodę JFD dla obrazów, opartą na skramblingu współczynników transformaty DCT. Wielokrotnie w literaturze, nazwa JFD odnosi się zarówno do grupy metod zgodnych z tą koncepcją, jak i do tej konkretnej metody. W tej metodzie, obraz jest dzielony na bloki 8×8 pikseli, a następnie poddawany jest transformacji DCT i kwantyzacji. Następnie, najistotniejsze perceptualnie współczynniki w blokach współczynników DCT są szyfrowane poprzez zmianę znaku, zgodnie z kluczem grupowym (kluczem scramblingu). Zaszyfrowane, istotne perceptualnie współczynniki są łączone z niezaszyfrowanymi, nieistotnymi perceptualnie współczynnikami tworząc zaszyfrowany obraz. Użytkownik dysponuje unikatowym kluczem deszyfrującym, który pozwala na odszyfrowanie jedynie podzbioru zaszyfrowanych współczynników, podczas gdy pozostałe współczynniki pozostają bez zmian. Kombinacja współczynników DCT, które po deszyfracji nadal mają odwrócony znak stanowi fingerprint. Niestety, niezauważalność fingerprintów nie jest zachowana. Możliwe jest zmniejszenie widoczności fingerprintów w obrazie, ale odbywa się to kosztem skuteczności detekcji fingerprintów. Ponadto, metoda charakteryzuje się niską odpornością na ataki zmowy. Szczegółowa analiza metody Kundura i Karthnika i propozycje usprawnień zostały przedstawiona w [71].

Następną, istotną metodą była metoda o nazwie Fingercasting zaproponowana przez Adelsbacha i innych [11]. Metoda Fingercasting jest uogólnieniem strumieniowego szyfru Chameleon dla danych dźwiękowych oraz obrazów i wykorzystuje osadzanie fingerprintów z rozpraszaniem widma. Istnieją dwie główne różnice między metodą Fingercasting, a metodą Chameleon. Po pierwsze, w metodzie Fingercasting, wpisy w tablicach odniesienia (ang. *look-up-table*) są losowane z rozkładem równomiernym. Po drugie, operacja XOR, występująca w metodzie Chameleon, została zastąpiona przez sumę w arytmetyce modularnej o wartości modułu większej niż 2. Odporność metody Fingercastingu na ataki zmowy wynika z odporności metody watermarkingu Cox'a z rozpraszaniem widma [20]. Dlatego też, Katzenbeisser i inni [72] zaproponowali ulepszenie tej metody o wykorzystanie kodów Tardos. W celu osiągnięcia wysokiej odporności na ataki zmowy, fingerprinty muszą być bardzo długie, co niestety prowadzi do zwiększenia rozmiaru tablicy odniesienia, a więc również rozmiaru kluczy deszyfrujących.

Lemma i inni [73] zaproponowali metodę JFD, w której współczynniki DC z bloków DCT obrazu są szyfrowane poprzez dodawanie losowego ciągu szyfrującego. Szyfrowanie współczynników DC powoduje znaczne pogorszenie jakości wizualnej zaszyfrowanego obrazu. Natomiast, deszyfrowanie współczynników DC polega na dodawaniu innego losowego ciągu, nazywanego ciągiem deszyfrującym. Ciąg deszyfrujący zawiera ciąg szyfrujący oraz ciąg fingerprintu danego użytkownika, co powoduje, że odszyfrowane dane zawierają osadzony fingerprint. W pracy [74] przedstawiono szczegółową analizę tej metody dla obrazów.

Wykazano, że w zaproponowanej implementacji często występuje przekroczenie zakresu wartości pikseli przy deszyfrowaniu oraz metoda Lemmy jest bardzo wrażliwa na ataki zmowy nawet przy zaledwie ok. 5 zmawiających się piratów. Lian i inni [74] zaproponowali szereg rozwiązań, które mają na celu m.in. zapobieganie przekraczaniu zakresów wartości danych, poprawę niezauważalności fingerprintów, szyfrowanie dodatkowych współczynników AC z bloków DCT, zwiększenie odporności na ataki zmowy. Po zastosowaniu zaproponowanych zmian, metoda jest odporna na ataki zmowy realizowane nawet przez ok. 50 piratów. W pracy [75] Lian i inni przedstawili analogiczną analizę metody Lemmy dla audio.

Celik, i inni [76] zaproponowali kolejną metodę JFD dla obrazów, która podobnie do metody Chameleon i metody Fingercasting jest oparta o tablice LUT (ang. *lookup-table*). W tej metodzie, podczas deszyfrowania wykorzystywane są operacje algebraiczne, które pozwalają na addytywne lub multiplikatywne osadzanie fingerprintów. Osadzone fingerprinty mogą być wydobyte poprzez detekcję niekoherentą. Ponadto, udoskonalono algorytm detekcji dla metod opartych o tablice LUT. Zaproponowany algorytm opiera się na konstruowaniu „podejrzanej tablicy LUT" z pirackiej kopii i charakteryzuje się mniejszą złożonością obliczeniową. Dodatkowo, metoda proponowana w [76] wykorzystuje właściwości maskowania perceptualnego ludzkiego wzroku w celu poprawy jakości oznakowanych kopii, przy utrzymaniu maksymalnej siły osadzania fingerprintów. Przy założeniu, że stosowane są fingerprinty ortogonalne, odporność na ataki zmowy jest tak wysoka, jak w metodzie z rozpraszaniem widma [20]. Niestety, autorzy podają [76], że możliwe jest wykorzystanie nadmiarowości w tablicach LUT do estymacji obrazu nieoznakowanego przy zmowie zaledwie 2 piratów.

Kolejną metodą JFD, która miała silny wpływ na rozwój nowych rozwiązań zaprezentowanych w kolejnych rozdziałach, była metoda Hillcast zaproponowana w [13, 29]. Metoda Hillcast wykorzystuje symetryczny szyfr blokowy będący uogólnionym szyfrem Hilla. Fragment obrazu w dziedzinie przestrzennej jest szyfrowany za pomocą klucza grupowego i obraz jest wysyłany do wszystkich użytkowników, za pomocą transmisji multicast. Zbiór unikatowych kluczy deszyfrujących zostaje wygenerowany na podstawie klucza grupowego oraz fingerprintów użytkowników. Różnice między kluczem grupowym, a kluczem deszyfrującym powodują zamierzone błędy w obrazie odszyfrowanym, które stanowią fingerprint użytkownika. Wyniki eksperymentów [13, 29] pokazały, że możliwe jest osiągnięcie niezauważalności osadzonych fingerprintów. Ponadto, przeprowadzone eksperymenty wykazały, że w przypadku ataku zmowy metoda jest w stanie identyfikować piratów z dużym prawdopodobieństwem, jeżeli liczba piratów w zmowie jest równa ok. 10 lub mniej. Jednakże, wykazano również, że istnieje stosunkowo duże ryzyko oskarżenia niewinnych użytkowników. Kolejną wadą tej metody jest fakt, iż fingerprint jest osadzany w wartościach pikseli obrazu, co powoduje, że jest on bardzo wrażliwy na ataki polegające na przetwarzaniu sygnałów, np. na kompresję. Ponadto, fingerprint jest osadzony jedynie we fragmencie obrazu, przez co jest on wrażliwy na ataki wycinania lub kadrowania (ang. *cropping attack*).

W ostatnich latach zaproponowano wiele metod JFD opartych na kwantyzacji wektorowej VQ (ang. *Vector Quantization*) [77, 78, 79, 80, 81]. Głównym powodem wykorzystania kwantyzacji wektorowej jest jej bardzo silna tolerancja na zakłócenia szumem. W tych metodach, obraz jest najpierw poddawany kompresji wykorzystującej VQ. W pierwszej metodzie [77, 78] obraz jest szyfrowany poprzez permutację oraz zastąpienie słów kodowych zgodnie ze statycznym drzewem kluczy (ang. *static key-trees*). W drugiej metodzie [77, 78] wykorzy-

stywane są dynamiczne drzewa kluczy oraz zrezygnowano z permutacji. Łączny fingerprinting i deszyfracja realizowany jest w oparciu o zbiór predefiniowanego zbioru drzew kluczy, które są unikatowe dla każdego użytkownika. Bezpieczeństwo metody Lin i innych [77] zostało zweryfikowane w pracy Li i innych [79]. Następnie, koncepcja ta została dalej rozwinięta w pracy [80]. Zaletą metody zaproponowanej przez Li i innych w [80] jest możliwość wydobycia fingerprintu z pirackiej kopii poprzez detekcję niekoherentną. Niestety, w badaniach wyżej wymienionych metod opartych o VQ nie rozpatrzono ich odporności na ataki zmowy.

Odmienne podejście, aczkolwiek bardzo podobne do koncepcji JFD, reprezentuje metoda Parness i Parviainen [82]. W tej metodzie, strona nadawcza generuje dwie, różnie oznakowane, kopie danych multimedialnych, które są przesyłane za pomocą dwóch transmisji multicast do wszystkich odbiorców. Każdy pakiet tych dwóch oznakowanych strumieni multicastowych jest szyfrowany innym kluczem. Użytkownicy posiadają unikatowe zbiory kluczy deszyfrujących, które są tworzone w taki sposób, że *l*-ty klucz pozwala na odszyfrowanie tylko jednego pakietu z *l*-tej pary pakietów, odebranej z obu transmisji multicast. Wówczas, unikatowa kombinacja odszyfrowanych pakietów z dwóch oznakowanych kopii stanowi fingerprint użytkownika. Niestety, metoda wymaga bardzo dużej liczby kluczy deszyfrujących, jej wymagania na dostępną przepustowość są dwukrotnie większe, ponieważ wykorzystywane są dwie transmisje multicast zamiast jednej, oraz strona dystrybucyjna realizuje osadzanie fingerprintów. Ponadto, metoda charakteryzuje się niską odpornością na ataki zmowy.

Xu i inni [83] zaproponowali metodę JFD operującą w pełni w skompresowanej domenie JPEG (ang. *JPEG compressed domain*) bez żadnego transkodowania, ani dekompresji. Dzięki temu, metoda jest wygodna w użyciu, gdyż chronione dane multimedialne rzadko są dostępne w postaci nieskompresowanej. Skompresowane dane są szyfrowane za pomocą zmienno-modularnego algorytmu szyfrowania opartego na mapowaniu przestrzennym. Unikatowy fingerprint użytkownika jest osadzany podczas deszyfrowania za pomocą unikatowego klucza danego użytkownika. Przeprowadzone eksperymenty [83] wykazały m.in., że osadzone fingerprinty są niezauważalne dla ludzkiego oka. Ponadto, wyniki eksperymentów pokazały, że w przypadku ataku zmowy metoda jest w stanie identyfikować piratów z dużym prawdopodobieństwem przy liczbie nie przekraczającej 12 piratów w zmowie.

Kolejna metoda JFD, zaproponowana przez Liana i innych [84], jest przeznaczona wyłącznie dla obrazów ruchomych. Po stronie nadawczej, dane wideo są szyfrowane poprzez szyfrowanie tylko wektorów ruchu MV (ang. *Motion Vector*). Zarówno deszyfrowanie, jak i osadzanie fingerprintu, są oparte na modyfikacjach wektorów ruchu i zostały połączone w jeden homogeniczny proces. Wówczas po stronie odbiorczej, wektory ruchu są deszyfrowane i jednocześnie jest w nich osadzany fingerprint. W pracy pokazano, że szyfrowanie samych wektorów ruchu pogarsza jakość wizualną zaszyfrowanego obrazu, ale w niewystarczającym stopniu i zaproponowano zastosowanie dodatkowego szyfrowania współczynników DCT. W celu osiągnięcia odporności na ataki zmowy, autorzy proponują wykorzystanie kodów odpornych na zmowę (ang. *collusion-resistant codes*) w roli fingerprintów. W pracy zaprezentowano wyniki badania odporności metody na ataki zmowy, jednakże zbadano jedynie zależność między skutecznością identyfikacji piratów, a jakością wizualną pirackiej kopii, natomiast nie podano liczebności zmowy.

Chen i inni [85] opracowali metodę JFD, w której liczba kluczy deszyfrujących jest mniejsza niż liczba użytkowników systemu. Autorzy zwracają uwagę na fakt, iż w metodach

JFD zazwyczaj U kluczy deszyfrujących należy wysłać poprzez bezpieczną transmisję unicast do U odbiorców. Dzięki użyciu metody podziału kluczy BIBD (ang. *Balanced Incomplete Block Design*), liczba kluczy deszyfrujących została zredukowana do rzędu $O(\sqrt{U})$. W tej metodzie, strona dystrybucyjna szyfruje dane za pomocą jednego klucza szyfrującego i zaszyfrowane dane wysyła multicastowo do U użytkowników. Następnie, klucz szyfrujący jest dekomponowany na zbiór υ pod-kluczy szyfrujących (ang. *scrambling subkeys*), gdzie $\upsilon < U$. W każdym z υ pod-kluczy szyfrujących osadzany jest inny watermark, co w rezultacie daje zbiór υ pod-kluczy deszyfrujących (ang. *descrambling subkeys*). Następnie, wszystkie pod-klucze są organizowane w υ zbiorów i wysyłane multicastowo do U użytkowników w taki sposób, że każdy użytkownik odbiera inną kombinację pod-kluczy, która podczas deszyfrowania osadzi różne watermarki. Kombinacja osadzonych watermarków stanowi fingerprint danego użytkownika. W tych badaniach nie rozpatrywano odporności na ataki zmowy.

4.4. Podsumowanie

W tym rozdziale, przedstawiono koncepcję łącznego fingerprintingu i deszyfracji, która charakteryzuje się wysoką skalowalnością zasobów obliczeniowych i sieciowych wraz z rosnącą liczbą użytkowników. Metody JFD, których geneza wiąże się z powstaniem kina cyfrowego, realizują osadzanie fingerprintów po stronie odbiorczej. Koncepcja opiera się na wykorzystaniu systemu kryptograficznego, w którym proces deszyfrowania celowo wprowadza błędy w obrazie. Wprowadzone błędy stanowią fingerprint użytkownika i wynikają one z różnic między kluczem szyfrującym i deszyfrującym. Ponieważ obraz jest szyfrowany jeden raz, niezależnie od liczby odbiorców, oraz fingerprint jest osadzany po stronie odbiorczej, osiągana jest wysoka wydajność konstrukcji. Ponieważ zaszyfrowane dane są wysyłane za pomocą jednej transmisji multicast do wszystkich odbiorców, osiągana jest wysoka wydajność dystrybucji.

W rozdziale wyjaśniono również, jak należy rozumieć bezpieczeństwo szyfrowania w kontekście metod JFD. Zdefiniowano dwa poziomy bezpieczeństwa i przedstawiono założenia dotyczące szyfrowania najistotniejszych perceptualnie elementów obrazu. Założenie to zostało przyjęte przy projektowaniu nowych metod JFD. W końcowej części rozdziału zamieszczono obszerny przegląd metod JFD, które miały wpływ na powstanie nowych metod JFD przedstawionych w niniejszej rozprawie.

5. METODA JFD OPARTA NA ILOCZYNIE MACIERZY

W tym rozdziale została zaproponowana nowa metoda łącznego fingerprintingu i deszyfracji oparta na iloczynie macierzy realizująca fingerprinting w dziedzinie transformaty DCT. Inspiracją do powstania tej metody była metoda Hillcast [13], która jest metodą JFD wykorzystującą uogólniony szyfr Hilla do szyfrowania obrazu w skali szarości i osadzania fingerprintów w dziedzinie przestrzennej, tj. w wartościach pikseli obrazu. Podczas implementacji i badania metody Hillcast [86] zaproponowano modyfikacje związane z wykorzystaniem arytmetyki modularnej, różnych typów fingerprintów, oraz transformaty Arnolda [87]. W wyniku tych badań, powstała zmodyfikowana metoda Hillcast [29, 88, 89, 90], która była pierwszą iteracją metody przedstawionej w tym rozdziale. Następnie, metoda została udoskonalona poprzez zmianę algorytmu szyfrowania i deszyfrowania w taki sposób, aby fingerprinting był realizowany w dziedzinie transformaty DCT. W rezultacie, udoskonalona metoda [30, 91] charakteryzuje się zwiększoną odpornością na próby usunięcia osadzonych fingerprintów. Następnie, zaproponowano mechanizm okresowej zamiany klucza [92] w celu uodpornienia tej metody na atak ze znanym tekstem otwartym. Nowa metoda JFD oparta na iloczynie macierzy [93], przedstawiona w tym rozdziale, powstała dzięki wnioskom wyciągniętym z wszystkich wcześniejszych badań i implementacji.

W kolejnych podrozdziałach przedstawiono opis proponowanej metody JFD opartej na iloczynie macierzy, składający się z następujących części: konstrukcja kluczy, szyfrowanie obrazu, realizowane po stronie nadawczej, łączny fingerprinting i deszyfracja, realizowane po stronie odbiorczej, oraz identyfikacja piratów realizowana przez stronę nadawczą lub zaufaną trzecią stronę. W ostatnim podrozdziale rozpatrzono słaby punkt szyfru Hilla, czyli wrażliwość na atak ze znanym tekstem otwartym, i zaproponowano mechanizm okresowej zamiany klucza, który zapewnia odporność proponowanej metody JFD na ten atak. Ponadto, przeprowadzono symulacyjne badania niezauważalności osadzonych fingerprintów, odporności na szum, odporności na kompresję, odporności na ataki zmowy oraz odporności na ataki zmowy połączone z kompresją. Wyniki przeprowadzonych badań zamieszczono w rozdziale 9.

5.1. Konstrukcja kluczy

Klucz szyfrujący $\mathbf{E}$ jest nieosobliwą macierzą kwadratową o rozmiarach $R \times R$. Elementy macierzy klucza szyfrującego $\mathbf{E}$ są losowo wybranymi liczbami całkowitymi. Podczas badania metody stosowano macierz $\mathbf{E}$, której elementy przyjmowały wartości z zakresu $\{0,1,2,3\}$ losowane zgodnie z rozkładem równomiernym. Aczkolwiek, macierz ta może być również binarna, tj. z elementami przyjmującymi wartości ze zbioru $\{0,1\}$. Należy zwrócić uwagę, że szyfrowanie nie jest realizowane w arytmetyce modularnej, zatem wartości zaszyfrowanych współczynników obrazu mogą rosnąć w bardzo szybkim tempie. Aby temu zapobiec, należy pomnożyć parzyste wiersze macierzy klucza szyfrującego $\mathbf{E}$ przez „–1”, aby nieparzyste wiersze zawierały tylko liczby dodatnie, a parzyste wiersze tylko liczby ujemne. Klucz szyfrujący $\mathbf{E}$ jest znany jedynie stronie nadawczej.

Fingerprint *u*-tego użytkownika o długości *M* zostaje zapisany w postaci macierzy $\mathbf{F}^{(u)}$ o rozmiarze $R \times R$, gdzie $u = 1,2,...,U$ oraz U jest liczbą użytkowników. Jeżeli długość fin-

gerprintu M jest mniejsza niż R^2, to macierz $\mathbf{F}^{(u)}$ musi zostać uzupełniona dodatkowymi wartościami „0”. W takim przypadku, zaleca się, aby ciąg stanowiący fingerprint był równomiernie rozproszony w całej macierzy $\mathbf{F}^{(u)}$, ponieważ będzie to miało wpływ na równomierne rozproszenie fingerprintu w zaszyfrowanym obrazie. Zbiór fingerprintów $\mathbf{F}^{(u)}$ jest znany jedynie stronie nadawczej.

Klucz deszyfrujący u-tego użytkownika $\mathbf{D}^{(u)}$ również jest macierzą o rozmiarze $R \times R$ i jest zależny od klucza szyfrującego i fingerprintu u-tego użytkownika w następujący sposób:

$$\mathbf{D}^{(u)} = \mathbf{E}^{-1} \cdot (\mathbf{I} + \alpha \cdot \mathbf{F}^{(u)}), \tag{5.1}$$

gdzie $\mathbf{D}^{(u)}$ jest macierzą klucza deszyfrującego u-tego użytkownika, $\mathbf{E}^{-1}$ jest macierzą odwrotną do macierzy klucza szyfrującego, $\mathbf{F}^{(u)}$ jest macierzą fingerprintu u-tego użytkownika, α jest współczynnikiem siły osadzania fingerprintu, oraz $\mathbf{I}$ jest macierzą jednostkową. Współczynnik siły osadzania fingerprintu α reprezentuje kompromis między niezauważalnością i odpornością osadzonych fingerprintów. Zmniejszanie wartości współczynnika α spowoduje, że fingerprint będzie coraz mniej widoczny w odszyfrowanym obrazie. Natomiast, zwiększanie wartości współczynnika α spowoduje, że fingerprint będzie bardziej odporny na próby usunięcia. Wartość współczynnika α powinna być wybrana w taki sposób, aby możliwe było osiągnięcie możliwie największej odporności fingerprintów przy jednoczesnym zachowaniu ich niezauważalności. Podczas badań, wartość współczynnika α została wybrana eksperymentalnie. Wygenerowane klucze deszyfrujące $\mathbf{D}^{(u)}$ należy dostarczyć poszczególnym użytkownikom w bezpieczny sposób.

5.2. Szyfrowanie

W celu zaszyfrowania obrazu kolorowego, należy go przekształcić do postaci macierzy obrazu $\mathbf{X}$, która zawiera współczynniki DCT wybrane do szyfrowania selektywnego. Najpierw, obraz kolorowy zostaje rozłożony na trzy podobrazy składowe zgodnie z modelem przestrzeni kolorów YCbCr, czyli na podobraz zawierający składową luminancji oraz dwa podobrazy zawierające składowe różnicowe chrominancji, z których jedna stanowi różnicę między luminancją, a kolorem niebieskim, a druga stanowi różnicę między luminancją, a kolorem czerwonym. Każda ze składowych zostaje podzielona na bloki o rozmiarze 8×8 pikseli. Następnie, każdy blok zostaje niezależnie poddany dwuwymiarowej dyskretnej transformacji kosinusowej DCT i przekształcony do postaci bloku 8×8 współczynników DCT. Można zaobserwować, że wstępne przetwarzanie obrazu posiada wiele cech wspólnych z transformacjami występującymi w znanych standardach kodowania MPEG oraz JPEG, co ułatwia integrację proponowanej metody z istniejącymi metodami kodowania obrazów. Szyfrowanie, a w rezultacie również osadzanie fingerprintów, jest realizowane tylko w składowej luminancji, ponieważ jest ona najbardziej istotna dla ludzkiego zmysłu wzroku oraz ma największą pojemność perceptualną. Niech $\mathbf{X}^{(b)}$ oznacza b-ty blok DCT składowej luminancji:

$$\mathbf{X}^{(b)} = \begin{bmatrix} x_{1,1}^{(b)} & x_{1,2}^{(b)} & \cdots & x_{1,8}^{(b)} \\ x_{2,1}^{(b)} & x_{2,2}^{(b)} & \cdots & x_{2,8}^{(b)} \\ \vdots & \vdots & \ddots & \vdots \\ x_{8,1}^{(b)} & x_{8,2}^{(b)} & \cdots & x_{8,8}^{(b)} \end{bmatrix}, \tag{5.2}$$

gdzie $x_{1,1}^{(b)}$, $x_{1,2}^{(b)}$, … $x_{8,8}^{(b)}$ są współczynnikami DCT *b*-tego bloku luminancji obrazu, $b = 1,2,...B$, a B jest liczbą wszystkich bloków DCT.

Szyfrowanie wszystkich współczynników DCT luminancji najlepiej zaszyfruje obraz. Jednakże, zgodnie z koncepcją szyfrowania selektywnego przedstawioną w rozdziale 4.2., w celu zmniejszenia złożoności obliczeniowej i czasu szyfrowania, zalecane jest zaszyfrowanie jedynie perceptualnie istotnych współczynników DCT. Należy zwrócić uwagę, że fingerprinty użytkowników będą osadzone tylko w zaszyfrowanych współczynnikach. W tym celu wybieranych jest $R \cdot (R-1)$ współczynników DCT składowej luminancji odpowiadających perceptualnie istotnym elementom obrazu, gdzie $R \times R$ jest rozmiarem macierzy klucza szyfrującego $\mathbf{E}$. Perceptualnie istotne współczynniki, odpowiadające widmowym składowym niskoczęstotliwościowym, znajdują się w lewym górnym rogu każdego bloku $\mathbf{X}^{(b)}$. Współczynniki wysokiego rzędu, które znajdują się w prawym dolnym rogu bloku, są wrażliwe na kompresję, filtrowanie dolnopasmowe, skalowanie, rozmycie, itp., zatem nie mogą być wybrane do szyfrowania, ponieważ spowodowałoby to, że osadzone fingerprinty byłyby wrażliwe na powyższe operacje przetwarzania obrazu. Należy zwrócić uwagę, iż współczynnik DC, znajdujący się w pierwszym rzędzie i pierwszej kolumnie każdego bloku, również nie może być wybrany do osadzania fingerprintu, ponieważ jego, nawet niewielka, zmiana w znaczący sposób pogarsza jakość perceptualną obrazu. W celu rozproszenia fingerprintu na całej powierzchni obrazu, współczynniki wybrane do szyfrowania powinny pochodzić z możliwie jak największej liczby bloków $\mathbf{X}^{(b)}$. Dla uproszczenia, bez utraty ogólności, niech liczba wszystkich bloków $B = R \cdot (R-1)$. Wówczas, należy wybrać po jednym współczynniku z każdego bloku DCT, na przykład można wybrać współczynniki w pierwszym rzędzie i drugiej kolumnie każdego bloku. W takim przypadku, wybrane współczynniki DCT tworzą macierz obrazu $\mathbf{X}$ o rozmiarze $(R-1) \times R$ w następujący sposób:

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,R} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,R} \\ \vdots & \vdots & \ddots & \vdots \\ x_{(R-1),1} & x_{(R-1),2} & \cdots & x_{(R-1),R} \end{bmatrix} = \begin{bmatrix} x_{1,2}^{(1)} & x_{1,2}^{(2)} & \cdots & x_{1,2}^{(R)} \\ x_{1,2}^{(R+1)} & x_{1,2}^{(R+2)} & \cdots & x_{1,2}^{(2\cdot R)} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1,2}^{((R-2)\cdot R+1)} & x_{1,2}^{((R-2)\cdot R+2)} & \cdots & x_{1,2}^{((R-1)\cdot R)} \end{bmatrix}. \tag{5.3}$$

Do opisu algorytmu wykorzystano przykład szyfrowania jednego współczynnika na blok, ale możliwe jest szyfrowanie większej liczby współczynników na każdy blok albo jednego współczynnika co kilka bloków. Jeżeli $B > R \cdot (R-1)$, to należy wybrać po jednym współczynniku z co $\lfloor B / R \cdot (R-1) \rfloor$-tego bloku DCT, gdzie $\lfloor a \rfloor$ oznacza zaokrąglenie liczby *a* w dół do najbliższej liczby całkowitej. Natomiast, jeżeli $B < R \cdot (R-1)$, to wówczas należy

wybrać po $\lceil R\cdot(R-1)/B\rceil$ współczynników z każdego bloku DCT, w kolejności zigzag (ang. *zigzag order*), gdzie $\lceil a\rceil$ oznacza zaokrąglenie liczby *a* w górę do najbliższej liczby całkowitej. Zamiast kolejności zigzag, można zastosować pseudolosowy algorytm wyboru współczynników, ale musi być on znany również stronie odbiorczej. Wybór współczynników w kolejności zigzag gwarantuje, że w pierwszej kolejności zostaną wybrane perceptualnie najistotniejsze współczynniki. Jeżeli istnieje potrzeba zaszyfrowania większej liczby współczynników niż $R\cdot(R-1)$, to należy uformować więcej macierzy obrazu i zrealizować szyfrowanie dla każdej macierzy niezależnie. Do dalszych obliczeń, wybrano przypadek szyfrowania jednego współczynnika na blok.

Wybrane współczynniki luminancji obrazu są szyfrowane w następujący sposób:

$$\mathbf{Y}=\mathbf{X}\cdot\mathbf{E}, \tag{5.4}$$

$$\mathbf{Y}=\begin{bmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,R} \\ y_{2,1} & y_{2,2} & \cdots & y_{2,R} \\ \vdots & \vdots & \ddots & \vdots \\ y_{(R-1),1} & y_{(R-1),2} & \cdots & y_{(R-1),R} \end{bmatrix}, \tag{5.5}$$

gdzie $\mathbf{X}$ jest macierzą obrazu zawierającą wybrane współczynniki DCT, $\mathbf{E}$ jest macierzą klucza szyfrującego, $\mathbf{Y}$ jest macierzą zaszyfrowanych współczynników DCT. Macierze $\mathbf{X}$ oraz $\mathbf{Y}$ mają rozmiar $(R-1)\times R$, natomiast macierz $\mathbf{E}$ ma rozmiar $R\times R$. Różnica w rozmiarach macierzy, tj. brak jednego wiersza, w połączeniu z okresową zamianą klucza, przedstawioną w rozdziale 5.10., daje odporność szyfru na atak ze znanym tekstem otwartym.

Następnie, zaszyfrowane współczynniki są umieszczane w swoich lokalizacjach w blokach DCT, z których uprzednio zostały wybrane. Wówczas, *b*-ty blok DCT składowej luminancji $\mathbf{Y}^{(b)}$ obrazu zaszyfrowanego ma następującą postać:

$$\mathbf{Y}^{(b)}=\begin{bmatrix} y_{1,1}^{(b)} & y_{1,2}^{(b)} & \cdots & y_{1,8}^{(b)} \\ y_{2,1}^{(b)} & y_{2,2}^{(b)} & \cdots & y_{2,8}^{(b)} \\ \vdots & \vdots & \ddots & \vdots \\ y_{8,1}^{(b)} & y_{8,2}^{(b)} & \cdots & y_{8,8}^{(b)} \end{bmatrix}=\begin{bmatrix} x_{1,1}^{(b)} & y_{g,h} & \cdots & x_{1,8}^{(b)} \\ x_{2,1}^{(b)} & x_{2,2}^{(b)} & \cdots & x_{2,8}^{(b)} \\ \vdots & \vdots & \ddots & \vdots \\ x_{8,1}^{(b)} & x_{8,2}^{(b)} & \cdots & x_{8,8}^{(b)} \end{bmatrix}, \tag{5.6}$$

gdzie $\mathbf{Y}^{(b)}$ jest *b*-tym blokiem zawierającym 63 niezaszyfrowane współczynniki $x_{1,1}^{(b)}$, $x_{1,2}^{(b)}$, … $x_{8,8}^{(b)}$ z *b*-tego bloku luminancji obrazu $\mathbf{X}^{(b)}$ oraz jeden zaszyfrowany współczynnik $y_{g,h}$ z *g*-tego wiersza i *h*-tej kolumny macierzy obrazu zaszyfrowanego $\mathbf{Y}$, $g=\lceil b/R\rceil$, $h=b-(\lceil b/R\rceil-1)\cdot R$, $b=1,2,...B$, B jest liczbą wszystkich bloków DCT, a $R\times R$ jest rozmiarem macierzy klucza szyfrującego $\mathbf{E}$.

Po szyfrowaniu, konieczne jest końcowe przekształcenie bloków DCT zaszyfrowanej luminancji oraz obu chrominancji, analogiczne do przekształceń w standardzie JPEG. Bloki DCT wszystkich składowych obrazu zostają poddane kwantyzacji DCT (kompresji), ustawio-

ne w kolejności zigzag, oraz poddane kompresji bezstratnej za pomocą kodowania metodą RLE (ang. *Run-Length Encoding*) i metodą Huffmana. Należy zwrócić uwagę, iż szyfrowanie jest przeprowadzane jednokrotnie, niezależnie od liczby użytkowników systemu. Zaszyfrowane dane są wysyłane do wszystkich użytkowników, za pomocą jednej transmisji rozgłoszeniowej typu mulicast.

5.3. Deszyfrowanie

W celu odszyfrowania danych po stronie odbiorczej, najpierw należy przeprowadzić przetwarzanie wstępne, które obejmuje dekodowanie RLE i Huffmana, odwrócenie kolejności zigzag oraz dekwantyzację DCT (dekompresji) odebranego obrazu. Następnie, strona odbiorcza musi odtworzyć macierz zaszyfrowanych współczynników $\mathbf{Y}$ z zaszyfrowanych bloków $\mathbf{Y}^{(b)}$ składowej luminancji obrazu. Po rekonstrukcji macierzy $\mathbf{Y}$, łączny fingerprinting i deszyfracja dla *u*-tego użytkownika realizowane jest w następujący sposób:

$$\mathbf{X}^{(u)} = \mathbf{Y} \cdot \mathbf{D}^{(u)}, \tag{5.7}$$

$$\mathbf{X}^{(u)} = \begin{bmatrix} x_{1,1}^{(u)} & x_{1,2}^{(u)} & \cdots & x_{1,R}^{(u)} \\ x_{2,1}^{(u)} & x_{2,2}^{(u)} & \cdots & x_{2,R}^{(u)} \\ \vdots & \vdots & \ddots & \vdots \\ x_{(R-1),1}^{(u)} & x_{(R-1),2}^{(u)} & \cdots & x_{(R-1),R}^{(u)} \end{bmatrix}, \tag{5.8}$$

gdzie $\mathbf{Y}$ jest macierzą zaszyfrowanych współczynników DCT, $\mathbf{D}^{(u)}$ jest macierzą klucza deszyfrującego *u*-tego użytkownika, $\mathbf{X}^{(u)}$ jest macierzą odszyfrowanych współczynników z fingerprintem *u*-tego użytkownika, $u = 1, 2, ...U$, a U jest liczbą wszystkich użytkowników. Macierze $\mathbf{X}^{(u)}$ oraz $\mathbf{Y}$ mają rozmiar $(R-1) \times R$, a macierz $\mathbf{E}$ ma rozmiar $R \times R$.

Następnie, odszyfrowane współczynniki są umieszczane w swoich lokalizacjach w blokach DCT, z których uprzednio zostały wybrane. Wówczas, *b*-ty blok DCT składowej luminancji $\mathbf{X}^{(u)(b)}$ obrazu odszyfrowanego z osadzonym fingerprintem *u*-tego użytkownika ma następującą postać:

$$\mathbf{X}^{(u)(b)} = \begin{bmatrix} x_{1,1}^{(u)(b)} & x_{1,2}^{(u)(b)} & \cdots & x_{1,8}^{(u)(b)} \\ x_{2,1}^{(u)(b)} & x_{2,2}^{(u)(b)} & \cdots & x_{2,8}^{(u)(b)} \\ \vdots & \vdots & \ddots & \vdots \\ x_{8,1}^{(u)(b)} & x_{8,2}^{(u)(b)} & \cdots & x_{8,8}^{(u)(b)} \end{bmatrix} = \begin{bmatrix} y_{1,1}^{(b)} & x_{g,h}^{(u)} & \cdots & y_{1,8}^{(b)} \\ y_{2,1}^{(b)} & y_{2,2}^{(b)} & \cdots & y_{2,8}^{(b)} \\ \vdots & \vdots & \ddots & \vdots \\ y_{8,1}^{(b)} & y_{8,2}^{(b)} & \cdots & y_{8,8}^{(b)} \end{bmatrix}, \tag{5.9}$$

gdzie $\mathbf{X}^{(u)(b)}$ jest *b*-tym blokiem zawierającym 63 współczynniki $y_{1,1}^{(b)}$, $y_{1,2}^{(b)}$, … $y_{8,8}^{(b)}$ z *b*-tego bloku luminancji odebranego obrazu $\mathbf{Y}^{(b)}$ oraz jeden odszyfrowany współczynnik $x_{g,h}^{(u)}$ z *g*-tego wiersza i *h*-tej kolumny macierzy obrazu odszyfrowanego $\mathbf{X}^{(u)}$, który zawiera część fingerprintu *u*-tego użytkownika, $g = \lceil b/R \rceil$, $h = b - (\lceil b/R \rceil - 1) \cdot R$, $b = 1, 2, ...B$, B jest licz-

bą wszystkich bloków DCT, $u = 1,2,...U$, U jest liczbą wszystkich użytkowników, a $R \times R$ jest rozmiarem macierzy klucza szyfrującego $\mathbf{E}$.

W ostatnim kroku, następuje przetwarzanie końcowe wszystkich bloków DCT odszyfrowanej składowej luminancji z osadzonym fingerprintem *u*-tego użytkownika oraz dwóch składowych chrominancji. Każdy blok zostaje niezależnie poddany dwuwymiarowej odwrotnej dyskretnej transformacji kosinusowej IDCT i przekształcony do postaci bloku 8×8 pikseli. Następnie, wszystkie bloki o rozmiarze 8×8 pikseli zostają połączone w celu odtworzenia trzech podobrazów składowych, zgodnie z modelem przestrzeni kolorów YCbCr. Ostatecznie, składowe, podobraz luminancji i dwa podobrazy chrominancji, zostają połączone w odszyfrowany obraz kolorowy z osadzonym fingerprintem *u*-tego użytkownika. Pomimo, że fingerprint został osadzony tylko w wybranych współczynnikach DCT, to po powrocie do domeny przestrzennej, tj. wartości pikseli obrazu, będzie on rozproszony we wszystkich pikselach każdego bloku, który zawiera odszyfrowany współczynnik DCT.

5.4. Identyfikacja piratów

Wydobycie fingerprintu z pirackiej kopii oraz identyfikacja nieuczciwych użytkowników, którzy wygenerowali tą kopię, są realizowane przez stronę dystrybucyjną lub zaufaną trzecią stronę. Zakłada się wykorzystanie detekcji koherentnej, ponieważ strona dystrybucyjna ma dostęp do oryginalnych danych oraz fingerprintów użytkowników.

W pierwszym kroku, należy przeprowadzić wydobycie fingerprintu obecnego w pirackiej kopii obrazu. W tym celu, strona dystrybucyjna musi odtworzyć macierz $\mathbf{X}$ z obrazu oryginalnego, czyli podzielić obraz na bloki 8×8 pikseli, przekształcić je w bloki 8×8 współczynników DCT, wybrać odpowiednie współczynniki luminancji i uformować je w macierz $\mathbf{X}$. Następnie, należy przeprowadzić analogiczne operacje dla pirackiego obrazu w celu otrzymania macierzy $\mathbf{X}_\mathrm{P}$ zawierającej wybrane współczynniki DCT luminancji pirackiego obrazu. Fingerprint, obecny w pirackiej kopii, jest wydobywany w następujący sposób:

$$\mathbf{F}_\mathrm{P} = \frac{1}{\alpha} \cdot (\mathbf{X}_\mathrm{P} - \mathbf{X}), \tag{5.10}$$

gdzie $\mathbf{F}_\mathrm{P}$ jest macierzą fingerprintu wydobytego z pirackiej kopii, $\mathbf{X}_\mathrm{P}$ jest macierzą obrazu pirackiego z obecnym fingerprintem pirata, $\mathbf{X}$ jest macierzą obrazu oryginalnego bez fingerprintu, natomiast α jest współczynnikiem siły osadzania fingerprintu.

W drugim kroku, należy obliczyć właściwą postać fingerprintów wszystkich użytkowników. Należy zwrócić uwagę, iż z równości (5.1), (5.4) i (5.7) wynika, że właściwy fingerprint *u*-tego użytkownika osadzony w obrazie $\mathbf{X}^{(u)}$ nie jest w postaci $\mathbf{F}^{(u)}$, jak w klasycznym modelu opisanym wzorem (2.2), ale jest iloczynem $\mathbf{X} \cdot \mathbf{F}^{(u)}$:

$$\mathbf{X}^{(u)} = \mathbf{Y} \cdot \mathbf{D}^{(u)} = \mathbf{X} \cdot \mathbf{E} \cdot \mathbf{E}^{-1} \cdot (\mathbf{I} + \alpha \cdot \mathbf{F}^{(u)}) = \mathbf{X} + \alpha \cdot \mathbf{X} \cdot \mathbf{F}^{(u)}. \tag{5.11}$$

Wówczas, właściwy fingerprint osadzony w kopii *u*-tego użytkownika jest równy:

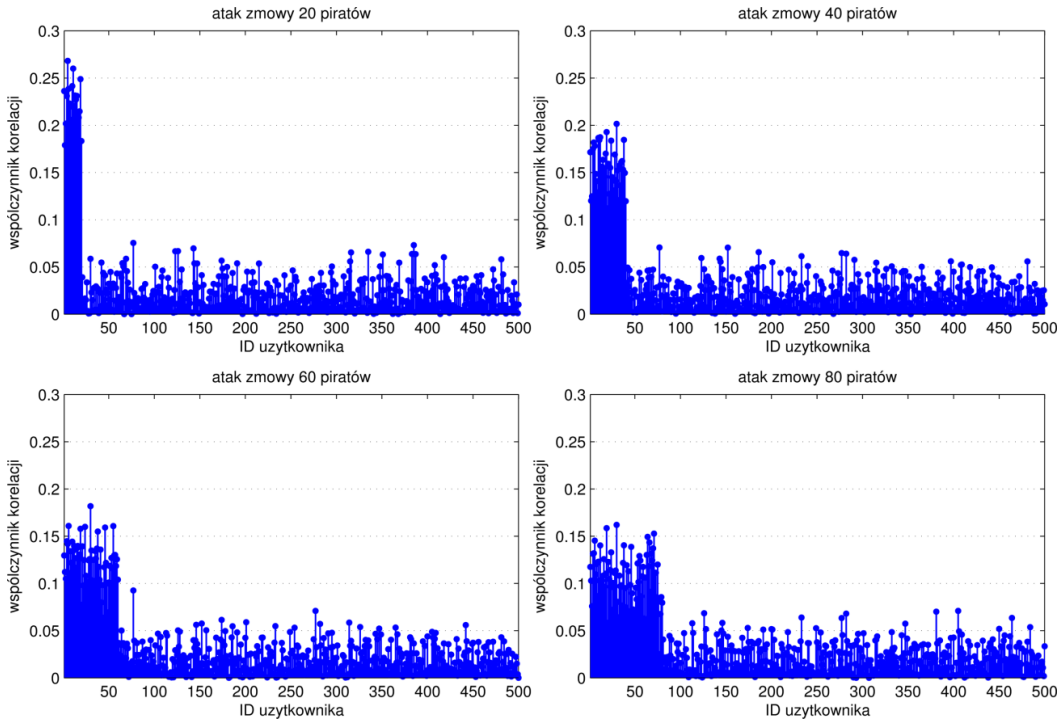
$$\mathbf{F}_{\text{emb}}^{(u)} = \mathbf{X} \cdot \mathbf{F}^{(u)}, \tag{5.12}$$

gdzie $\mathbf{F}_{\text{emb}}^{(u)}$ jest macierzą właściwego fingerprintu osadzonego w kopii *u*-tego użytkownika, $\mathbf{X}$ jest macierzą obrazu oryginalnego, $\mathbf{F}^{(u)}$ jest macierzą fingerprintu *u*-tego użytkownika, $u = 1, 2, ...U$, natomiast U jest liczbą wszystkich użytkowników.

W ostatnim kroku realizowana jest analiza korelacyjna. Obliczany jest zbiór U współczynników korelacji skrośnej $\rho^{(u)}$, $u = 1, 2, ...U$, między fingerprintem wydobytym z pirackiej kopii $\mathbf{F}_{\text{P}}$ oraz fingerprintem każdego użytkownika $\mathbf{F}_{\text{emb}}^{(u)}$:

$$\rho^{(u)} = \left| \frac{\sum_{r_1=1}^{R-1} \sum_{r_2=1}^{R} (f_{r_1, r_2\,\text{P}} - \overline{\mathbf{F}_{\text{P}}})(f_{r_1, r_2\,\text{emb}}^{(u)} - \overline{\mathbf{F}_{\text{emb}}^{(u)}})}{\sqrt{\left(\sum_{r_1=1}^{R-1} \sum_{r_2=1}^{R} (f_{r_1, r_2\,\text{P}} - \overline{\mathbf{F}_{\text{P}}})^2 \right) \left(\sum_{r_1=1}^{R-1} \sum_{r_2=1}^{R} (f_{r_1, r_2\,\text{emb}}^{(u)} - \overline{\mathbf{F}_{\text{emb}}^{(u)}})^2 \right)}} \right|, \tag{5.13}$$

gdzie $\rho^{(u)}$ jest wartością bezwzględną współczynnika korelacji macierzy $\mathbf{F}_{\text{P}}$ i $\mathbf{F}_{\text{emb}}^{(u)}$, $\overline{\mathbf{F}_{\text{P}}}$ jest wartością średnią wszystkich składowych macierzy $\mathbf{F}_{\text{P}}$, a $\overline{\mathbf{F}_{\text{emb}}^{(u)}}$ jest wartością średnią wszystkich składowych macierzy $\mathbf{F}_{\text{emb}}^{(u)}$. Następnie, współczynniki korelacji $\rho^{(u)}$ są porównywane z progiem detekcji t_{D}. Jeżeli współczynnik korelacji $\rho^{(u)}$ dla fingerprintu *u*-tego użytkownika jest większy od progu detekcji t_{D}, to wówczas *u*-ty użytkownik zostaje wykryty jako pirat.



Rys. 5.1. Przykłady wartości współczynników korelacji między fingerprintem wydobytym z pirackiej kopii oraz fingerprintem każdego użytkownika dla metody opartej na iloczynie macierzy (zmówieni piraci mają początkowe wartości identyfikatorów), opracowanie własne.

Na rysunku 5.1. przedstawiono przykładowe wartości współczynników korelacji dla fingerprintu wydobytego z pirackiej kopii oraz fingerprintów poszczególnych użytkowników. Przedstawiono cztery przypadki, w których piracka kopia została utworzona przez 20, 40, 60 lub 80 piratów. Dla ułatwienia analizy wykresów, w każdym ataku zmowy, zmawiający się piraci mieli początkowe wartości identyfikatorów. Można zaobserwować, że wartości unormowanych współczynników korelacji dla fingerprintów piratów są znacznie większe niż wartości współczynników dla fingerprintów niewinnych użytkowników, co umożliwia identyfikację piratów.

5.5. Okresowa zmiana klucza

Szyfr Hilla jest klasycznym symetrycznym szyfrem blokowym wykorzystującym przekształcenia macierzowe. Szyfr ten nie jest uznawany za bezpieczny, gdyż po zebraniu R różnych par wektorów tekstu otwartego i odpowiadających im wektorów szyfrogramów, gdzie $R \times R$ to rozmiar macierzy klucza, szyfr Hilla może być złamany w ataku kryptograficznym ze znanym tekstem otwartym. Jego podatność na kryptoanalizę sprawiła, że uznano go za nieprzydatny do praktycznych zastosowań, pomimo jego istotnych zalet. Szyfr ten jest bardzo prosty i szybki ze względu na korzystanie jedynie z mnożenia i odwracania macierzy w celu szyfrowania i deszyfrowania, a przy tym bardzo skutecznie ukrywa częstości występowania symboli w tekście otwartym. W literaturze można znaleźć prace przedstawiające próby poprawy bezpieczeństwa szyfru Hilla, które można podzielić na dwa następujące podejścia. Pierwsze podejście polega na zaprojektowaniu bezpiecznego mechanizmu wymiany klucza [94], natomiast drugie podejście polega na opracowaniu nieliniowej wersji szyfru Hilla [95]. W tym rozdziale zaproponowano rozwiązanie należące do pierwszej grupy.

Proponowana metoda JFD wykorzystuje szyfr będący uogólnionym szyfrem Hilla, zatem oferuje taki sam poziom bezpieczeństwa, który opiera się na rozmiarze macierzy szyfrowanej $\mathbf{X}$ oraz macierzy klucza $\mathbf{E}$. Im większy jest rozmiar macierzy szyfrowanej $\mathbf{X}$, tym zaszyfrowane współczynniki są silniej rozkorelowane. Natomiast, im większy jest rozmiar macierzy klucza $\mathbf{E}$, tym atak krytograficzny jest bardziej czasochłonny. Jak wspomniano wcześniej, słabą stroną szyfru Hilla jest jego wrażliwość na atak ze znanym tekstem otwartym. W celu uodpornienia metody na to zagrożenie, strona dystrybucyjna powinna zmieniać klucz $\mathbf{E}$ po zaszyfrowaniu każdej macierzy $\mathbf{X}$ oraz dostarczać nowe klucze deszyfrujące $\mathbf{D}^{(u)}$ wszystkim użytkownikom. Jednakże, dystrybucja strumienia kolejnych nowych kluczy deszyfrujących, indywidualnie do każdego użytkownika, jest bardzo niepraktyczna, zwłaszcza przy bardzo dużej liczbie kluczy. Aby rozwiązać ten problem, zaproponowano bezpieczny mechanizm okresowej zamiany klucza, w którym dodatkowa informacja dotycząca zamiany klucza jest wysyłana jednokrotnie wraz z dostarczeniem pierwszego klucza deszyfrującego.

Na początku transmisji, generowany jest pierwszy klucz szyfrujący $\mathbf{E}_1$, który jest stosowany do zaszyfrowania tylko pierwszej macierzy obrazu, zgodnie ze wzorem (5.4). Dodatkowo, strona dystrybucyjna wybiera generator liczb pseudolosowych PRNG (and. *Pseudo-Random Number Generator*) i jego ziarno (ang. *seed*). Generator PRNG musi być znany stronie odbiorczej. Pierwsze klucze deszyfrujące $\mathbf{D}_1^{(u)}$ są konstruowane zgodnie ze wzorem (5.1) i wysyłane do wszystkich użytkowników, razem z ziarnem generatora, za pomocą bezpiecz-

nych połączeń unicast, gdzie $u = 1,2,...,U$, oraz U jest liczbą użytkowników. W celu utworzenia l-tego klucza szyfrującego, gdzie $l = 2,3,...$, strona dystrybucyjna musi wygenerować losową macierz permutacji $\mathbf{P}_l$ o rozmiarze $R \times R$ za pomocą generatora PRNG i ustalonego ziarna. Następnie, l-ty klucz szyfrujący jest generowany zgodnie z poniższym wzorem:

$$\mathbf{E}_l = \mathbf{E}_{l-1} \cdot \mathbf{P}_l, \tag{5.16}$$

gdzie $\mathbf{P}_l$ jest l-tą macierzą permutacji, $\mathbf{E}_{l-1}$ jest poprzednim kluczem szyfrującym, $\mathbf{E}_l$ jest nowym kluczem szyfrującym, $\mathbf{E}_1$ jest pierwszym kluczem szyfrującym, oraz $l = 1,2,...$. Wszystkie macierze mają rozmiar $R \times R$. Następnie, l-ta macierz obrazu zostaje zaszyfrowana za pomocą l-tego klucza szyfrującego $\mathbf{E}_l$, zgodnie z wzorem (5.4).

Każda strona odbiorcza otrzymuje swój pierwszy klucz deszyfrujący $\mathbf{D}_1^{(u)}$ oraz ziarno generatora PRNG. Klucz deszyfrujący $\mathbf{D}_1^{(u)}$ służy do odszyfrowania tylko pierwszej macierzy zaszyfrowanego obrazu, zgodnie z wzorem (5.7). Odebrane ziarno jest niezbędne do zsynchronizowania generatora PRNG po stronie dystrybucyjnej oraz po stronie wszystkich odbiorców. W celu utworzenia l-tego klucza szyfrującego, gdzie $l = 1,2,...$, strona odbiorcza musi wygenerować losową macierz permutacji $\mathbf{P}_l$ o rozmiarze $R \times R$ za pomocą generatora PRNG i odebranego ziarna. Następnie, l-ty klucz deszyfrujący u-tego użytkownika jest generowany zgodnie z poniższym wzorem:

$$\mathbf{D}_l^{(u)} = \mathbf{P}_l^{-1} \cdot \mathbf{D}_{l-1}^{(u)}, \tag{5.17}$$

gdzie $\mathbf{P}_l^{-1}$ jest macierzą odwrotną do l-tej macierzy permutacji, $\mathbf{D}_{l-1}^{(u)}$ jest poprzednim kluczem deszyfrującym u-tego użytkownika, $\mathbf{D}_1^{(u)}$ jest nowym kluczem deszyfrującym u-tego użytkownika, $\mathbf{D}_1^{(u)}$ jest pierwszym kluczem deszyfrującym u-tego użytkownika, $l = 1,2,...$, $u = 1,2,...,U$, oraz U jest liczbą użytkowników. Wszystkie macierze mają rozmiar $R \times R$. Następnie, l-ta macierz zaszyfrowanego obrazu zostaje odszyfrowana za pomocą l-tego klucza deszyfrującego $\mathbf{D}_1^{(u)}$, zgodnie z wzorem (5.7)

Przedstawiona okresowa zmiana klucza nie powoduje żadnych zmian w strukturze kluczy deszyfrujących i jest transparentna z punktu widzenia fingerprintingu realizowanego w proponowanej metodzie JFD. Ponadto, przedstawiony mechanizm okresowej zmiany klucza nie wymaga wysyłania nowych kluczy deszyfrujących do każdej macierzy zaszyfrowanego obrazu. Zamiast tego, wysyłany jest jedynie pierwszy klucz deszyfrujący i ziarno generatora liczb pseudolosowych, a strona odbiorcza sama generuje kolejne klucze. Dodatkowa informacja, którą jest ziarno, jest wysyłana jednokrotnie, przy rozpoczęciu transmisji.

5.6. Podsumowanie

W tym rozdziale przedstawiono nową metodę łącznego fingerprintingu i deszyfracji opartą na iloczynie macierzy, dla obrazów kolorowych. Proponowana metoda wykorzystuje szyfr, będący uogólnionym szyfrem Hilla, do szyfrowania składowej luminancji obrazu. W metodzie została wykorzystana koncepcja szyfrowania selektywnego, tzn. obraz nie jest szyfrowany w całości, ale szyfrowane są jedynie wybrane, istotne perceptualnie, współczynniki DCT. Zarówno szyfrowanie, jak i fingerprinting są realizowane w dziedzinie dyskretnej transformaty kosinusowej, dzięki czemu metoda charakteryzuje się wysoką odpornością na wiele ataków polegających na przetwarzaniu sygnału obrazu. Ponadto, w dziedzinie przestrzennej fingerprinty są rozproszone na całej powierzchni obrazu, dzięki czemu nie mogą być w łatwy sposób wycięte z obrazu. Proponowana metoda charakteryzuje się wysoką wydajnością konstrukcji, ponieważ szyfrowanie jest realizowane jednokrotnie, niezależnie od liczby użytkowników, a osadzanie fingerprintu jest realizowane po stronie odbiorczej. Proponowana metoda charakteryzuje się również wysoką wydajnością dystrybucji, gdyż wystarczy jedna transmisja multicast w celu dostarczenia oznakowanych kopii do wszystkich użytkowników. Na końcu rozdziału przedstawiono bezpieczną okresową zamianę kluczy, której celem jest uodpornienie stosowanego szyfru na atak ze znanym tekstem otwartym.

6. PODSTAWY RACHUNKU KWATERNIONOWEGO

Kwaterniony są rozszerzeniem ciała liczb zespolonych opracowanym przez Williama Rowana Hamiltona w 1843 roku. W algebrze kwaternionów, podstawową jednostką jest kwaternion składający się z 4 skalarnych współczynników. W ciele kwaternionów zdefiniowano działania addytywne oraz multiplikatywne, jednakże jest to ciało nieprzemienne ze względu na mnożenie. Głównym przeznaczeniem kwaternionów jest matematyczny opis transformacji w przestrzeni trójwymiarowej za pomocą czterowymiarowych liczb.

W tym rozdziale zostały przedstawione kwaterniony w postaci kartezjańskiej oraz dwuczłonowej. Następnie, zdefiniowano sumę i iloczyn kwaternionów, a także sprzężenie, odwrotność oraz normę kwaternionu. Na końcu rozdziału przedstawiono możliwości wykorzystania kwaternionów do opisu przekształceń w przestrzeni trójwymiarowej. Powyższe operacje zostały wykorzystane w nowych metodach JFD opisany w kolejnych rozdziałach. Więcej informacji na temat kwaternionów można znaleźć w [14, 15].

6.1. Algebra kwaternionów

Kwaternion jest 4-wymiarową liczbą, posiadającą 1 składową rzeczywistą i 3 składowe urojone. Każdy z trzech urojonych wymiarów jest prostopadły do dwóch pozostałych i posiada własną jednostkę urojoną. Kwaternion może być przedstawiony w następującej, kartezjańskiej postaci:

$$q = q_w + i \cdot q_x + j \cdot q_y + k \cdot q_z , \tag{6.1}$$

gdzie q jest kwaternionem, q_w jest składową rzeczywistą, q_x, q_y, q_z, są składowymi urojonymi, natomiast i, j, k są jednostkami urojonymi, które charakteryzują się następującymi właściwościami:

$$i \cdot i = j \cdot j = k \cdot k = i \cdot j \cdot k = -1, \tag{6.2}$$

$$i \cdot j = k , \tag{6.3}$$

$$j \cdot i = -k , \tag{6.4}$$

$$j \cdot k = i , \tag{6.5}$$

$$k \cdot j = -i , \tag{6.6}$$

$$k \cdot i = j , \tag{6.7}$$

$$i \cdot k = -j . \tag{6.8}$$

Alternatywnie, kwaternion można przedstawić w postaci dwuczłonowej, składającej się z części skalarnej (rzeczywistej) i części wektorowej (zawierającej składowe urojone):

$$q = (q_w, \vec{q}_v) = (q_w, \begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix}) . \tag{6.9}$$

Suma kwaternionów jest obliczana poprzez dodawanie odpowiadających sobie współczynników kwaternionów, podobnie jak w przypadku liczb zespolonych. Wówczas, suma kwaternionów p i q wynosi:

$$\begin{aligned} p+q &= (p_w + i\cdot p_x + j\cdot p_y + k\cdot p_z) + (q_w + i\cdot q_x + j\cdot q_y + k\cdot q_z) = \\ &= (p_w + q_w) + i\cdot(p_x + q_x) + j\cdot(p_y + q_y) + k\cdot(p_z + q_z) \end{aligned}, \tag{6.10}$$

gdzie p_w, p_x, p_y, p_z są współczynnikami kwaternionu p, q_w, q_x, q_y, q_z są współczynnikami kwaternionu q, a i, j, k są jednostkami urojonymi. Warto zwrócić uwagę, że dla dodawania kwaternionów spełniona jest zasada łączności, tj.:

$$(q_1 + q_2) + q_3 = q_1 + (q_2 + q_3), \tag{6.11}$$

gdzie q_1, q_2, q_3 są kwaternionami. Alternatywnie, sumę kwaternionów można przedstawić w ich dwuczłonowej postaci:

$$p+q = (p_w + q_w, \vec{p}_v + \vec{q}_v) = (p_w + q_w, \begin{bmatrix} p_x + q_x \\ p_y + q_y \\ p_z + q_z \end{bmatrix}). \tag{6.12}$$

Różnica kwaternionów p i q jest obliczana jako suma kwaternionu p i kwaternionu q z przeciwnym znakiem.

Iloczyn kwaternionów jest określony przez właściwości jednostek urojonych i, j, k, które zostały przedstawione wzorami od (6.2) do (6.8). Należy zwrócić uwagę, że z właściwości jednostek urojonych wynika, że mnożenie kwaternionów nie jest przemienne. Iloczyn kwaternionów p i q jest również kwaternionem i obliczany jest w następujący sposób:

$$\begin{aligned} p\cdot q &= (p_w + i\cdot p_x + j\cdot p_y + k\cdot p_z)\cdot(q_w + i\cdot q_x + j\cdot q_y + k\cdot q_z) = \\ &= (p_w q_w - p_x q_x - p_y q_y - p_z q_z) + i\cdot(p_w q_x + p_x q_w + p_y q_z - p_z q_y) + \\ &+ j\cdot(p_w q_y + p_y q_w + p_z q_x - p_x q_z) + k\cdot(p_w q_z + p_z q_w + p_x q_y - p_y q_x) \end{aligned}, \tag{6.13}$$

gdzie p_w, p_x, p_y, p_z są współczynnikami kwaternionu p, q_w, q_x, q_y, q_z są współczynnikami kwaternionu q, a i, j, k są jednostkami urojonymi. Należy zwrócić uwagę, że dla mnożenia kwaternionów spełniona jest zasada łączności, tj.:

$$(q_1 \cdot q_2)\cdot q_3 = q_1 \cdot (q_2 \cdot q_3), \tag{6.14}$$

gdzie q_1, q_2, q_3 są kwaternionami. Alternatywnie, iloczyn kwaternionów można przedstawić w ich dwuczłonowej postaci:

$$p \cdot q = (p_w \cdot q_w - \vec{p}_v \bullet \vec{q}_v,\ p_w \cdot \vec{q}_v + q_w \cdot \vec{p}_v + \vec{p}_v \times \vec{q}_v) = \\ = (p_w q_w - p_x q_x - p_y q_y - p_z q_z, \begin{bmatrix} p_w q_x + p_x q_w + p_y q_z - p_z q_y \\ p_w q_y + p_y q_w + p_z q_x - p_x q_z \\ p_w q_z + p_z q_w + p_x q_y - p_y q_x \end{bmatrix}). \tag{6.15}$$

gdzie $\bullet$ oznacza iloczyn skalarny (ang. *vector dot product*):

$$\vec{p}_v \bullet \vec{q}_v = p_x q_x + p_y q_y + p_z q_z, \tag{6.16}$$

natomiast $\times$ oznacza iloczyn wektorowy (ang. *vector cross product*):

$$\vec{p}_v \times \vec{q}_v = \det\left(\begin{bmatrix} i & j & k \\ p_x & p_y & p_z \\ q_x & q_y & q_z \end{bmatrix} \right), \tag{6.17}$$

gdzie det(·) oznacza wyznacznik macierzy. Iloraz kwaternionów p i q jest obliczany jako iloczyn kwaternionu p oraz odwrotności kwaternionu q, która jest zdefiniowana wzorem (6.22).

Sprzężenie kwaternionu q jest kwaternionem q^* o takiej samej składowej rzeczywistej co kwaternion q, ale o przeciwnym znaku każdej składowej urojonej:

$$q^* = q_w - i \cdot q_x - j \cdot q_y - k \cdot q_z. \tag{6.18}$$

Ponadto, sprzężenie kwaternionu charakteryzuje się następującymi właściwościami:

$$(p \cdot q)^* = q^* \cdot p^*, \tag{6.19}$$

$$q \cdot q^* = q_w^2 + q_x^2 + q_y^2 + q_z^2. \tag{6.20}$$

Norma kwaternionu q jest utożsamiana z jego modułem i oblicza się ją następująco:

$$\|q\| = \sqrt{q \cdot q^*} = \sqrt{q_w^2 + q_x^2 + q_y^2 + q_z^2}. \tag{6.21}$$

W celu znormalizowania kwaternionu q, należy podzielić każdą jego składową q_w, q_x, q_y, q_z przez normę tego kwaternionu $\|q\|$. Norma kwaternionu znormalizowanego jest równa 1.

Odwrotność kwaternionu q zdefiniowana jest następujący sposób:

$$q^{-1} = \frac{q^*}{\|q\|^2} = \frac{q_w - i \cdot q_x - j \cdot q_y - k \cdot q_z}{q_w^2 + q_x^2 + q_y^2 + q_z^2}. \tag{6.22}$$

Z powyższej definicji wynika, że jeżeli kwaternion q jest znormalizowany, to wtedy:

$$q^{-1} = q^{*}. \tag{6.23}$$

Ponadto, dla odwrotności iloczynu dwóch kwaternionów zawsze zachodzi:

$$(p \cdot q)^{-1} = q^{-1} \cdot p^{-1}. \tag{6.24}$$

6.2. Przekształcenia punktu w przestrzeni trójwymiarowej

Głównym zastosowaniem kwaternionów jest matematyczny opis przekształceń w przestrzeni 3-wymiarowej, a w szczególności rotacji (obrotu) punktów i wektorów. Opisywanie rotacji 3-wymiarowej za pomocą trzech wartości skalarnych, np. za pomocą kątów Eulera, jest problematyczne, ponieważ takie przekształcenia są nieliniowe, posiadają osobliwości i są trudne w łączeniu [14, 15]. W celu ominięcia tych problemów, przestrzeń 3-wymiarowa nie jest modelowana bezpośrednio, ale jest odwzorowywana w przestrzeni 4-wymiarowej. W celu zrozumienia tego zjawiska, w analogiczny sposób można rozpatrywać 2-wymiarową mapę Świata. Nie jest możliwe przedstawienie mapy Świata na 2-wymiarowej mapie z jednoczesnym wiernym odwzorowaniem zarówno kątów, jak i odległości. Jednakże, kiedy 2-wymiarowa mapa zostanie owinięta wokół 3-wymiarowej sfery, wierne odwzorowanie staje się możliwe, a operacje 2-wymiarowe stają się liniowe w nowej przestrzeni. Podobnie, przekształcenia w przestrzeni 3-wymiarowej stają się liniowe po odwzorowaniu ich w 4-wymiarowej przestrzeni kwaternionów [14, 15].

Kwaterniony mogą być wykorzystywane do opisu wielu różnych przekształceń w przestrzeni 3-wymiarowej, m.in. do obliczania wyniku rotacji, odbicia, skalowania lub przesunięcia punktu. Punkt (lub wektor) w przestrzeni 3-wymiarowej jest reprezentowany przez kwaternion, którego składowa rzeczywista jest równa 0, natomiast składowe urojone stanowią współrzędne tego punktu (lub wektora). Poniżej, przedstawiono zapis powyższych przekształceń za pomocą rachunku kwaternionowego [15]. Oznaczenia we wzorach od (6.25) do (6.29) są następujące: p_{IN} jest kwaternionem reprezentującym punkt początkowy przed przekształceniem, q jest kwaternionem reprezentującym punkt lub wektor, względem którego jest realizowane dane przekształcenie, natomiast p_{OUT} jest kwaternionem reprezentującym punkt wynikowy po przekształceniu.

Skalowanie kwaternionu p_{IN} o współczynnik skalarny a jest obliczane zgodnie z:

$$p_{\mathrm{OUT}} = a \cdot p_{\mathrm{IN}}. \tag{6.25}$$

Przesunięcie kwaternionu p_{IN} o wektor q jest obliczane w następujący sposób:

$$p_{\mathrm{OUT}} = p_{\mathrm{IN}} + q. \tag{6.26}$$

Odbicie kwaternionu p_{IN} poprzez kwaternion q jest obliczane w następujący sposób:

$$p_{\mathrm{OUT}} = q \cdot p_{\mathrm{IN}} \cdot q\,. \tag{6.27}$$

Rotacja kwaternionu p_{IN} wokół kwaternionu q jest obliczana z wykorzystaniem iloczynu obustronnego (ang. *sandwich product*):

$$p_{\mathrm{OUT}} = q \cdot p_{\mathrm{IN}} \cdot q^{-1}, \tag{6.28}$$

lub, jeżeli kwaternion q jest znormalizowany:

$$p_{\mathrm{OUT}} = q \cdot p_{\mathrm{IN}} \cdot q^{*}. \tag{6.29}$$

Możliwe jest wyszczególnienie dwóch sposobów rotacji. Po pierwsze, punkt reprezentowany przez kwaternion p_{IN} może być obracany względem innego punktu w przestrzeni 3-wymiarowej. Wówczas, kwaternion q reprezentuje ten punkt, a więc jego składowa rzeczywista jest równa 0, a składowe urojone stanowią współrzędne punktu. Po drugie, punkt reprezentowany przez kwaternion p_{IN} może być obracany o zadany kąt względem zadanej płaszczyzny. Wówczas, kwaternion q jest wyznaczany w następujący sposób:

$$q = \cos(\gamma/2) + i \cdot (a_x \cdot \sin(\gamma/2)) + j \cdot (a_y \cdot \sin(\gamma/2)) + k \cdot (a_z \cdot \sin(\gamma/2))\,. \tag{6.30}$$

gdzie γ jest kątem rotacji, a $[a_x, a_y, a_z]$ jest wektorem opisującym płaszczyznę rotacji.

Istotną zaletą rotacji kwaternionowych jest łatwość ich łączenia. Jeżeli kwaternion p_{IN} jest poddawany rotacji względem kwaternionu q_1, a następnie kolejnej rotacji względem q_2, to możliwe jest połączenie tych dwóch rotacji w jedną rotację łączną względem kwaternionu $q_2 \cdot q_1$, co przedstawiono w poniższym wzorze:

$$p_{\mathrm{OUT}} = q_2 \cdot (q_1 \cdot p_{\mathrm{IN}} \cdot q_1^{-1}) \cdot q_2^{-1} = (q_2 \cdot q_1) \cdot p_{\mathrm{IN}} \cdot (q_1^{-1} \cdot q_2^{-1}) = (q_2 \cdot q_1) \cdot p_{\mathrm{IN}} \cdot (q_2 \cdot q_1)^{-1}. \tag{6.31}$$

6.3. Podsumowanie

W tym rozdziale zostały przedstawione podstawy rachunku kwaternionowego, w zakresie niezbędnym dla zrozumienia metod JFD opisanych w kolejnych rozdziałach. W ramach tego przedstawienia, zdefiniowano kwaternion oraz jego postać kartezjańską i dwuczłonową. Następnie, zdefiniowano sumę i iloczyn kwaternionów, a także sprzężenie, odwrotność oraz normę kwaternionu. Na końcu rozdziału opisano możliwości wykorzystania kwaternionów do opisu przekształceń punktów w przestrzeni trójwymiarowej. Wyszczególniono takie przekształcenia jak rotacja, odbicie, skalowanie i przesunięcie punktu lub wektora, reprezentowanego przez kwaternion.

Przedstawione podstawy rachunku kwaternionowego zostały wykorzystane w zaproponowanych nowych metodach JFD opartych o kwaternionową reprezentację obrazów kolorowych, których opis znajduje się dwóch kolejnych rozdziałach. W tych metodach, do elementów obrazu, kluczy oraz fingerprintów zastosowano reprezentację kwaternionową, a algorytm szyfrowania i deszyfrowania oparty jest o rotację kwaternionową.

7. METODA JFD OPARTA NA ROTACJI KWATERNIONÓW Z CQC

W tym rozdziale została zaproponowana nowa metoda łącznego fingerprintingu i deszyfracji oparta na rotacji kwaternionów i realizująca fingerprinting w dziedzinie transformaty DCT. Inspiracją do powstania tej metody była metoda przedstawiona w [37, 96, 97], w której po raz pierwszy wykorzystano rachunek kwaternionowy do realizacji fingerprintingu. W tej metodzie [37, 96, 97] fingerprinting był przedstawiony jako dodatkowa funkcjonalność systemu szyfrowania kwaternionowego, w którym kwaterniony danych są obracane w 3-wymiarowej przestrzeni wokół kwaternionu klucza. Po stronie odbiorczej, podczas deszyfrowania, w obrazie pojawiają się pewne niewidoczne błędy w obrazie, które są unikatowe dla każdej pary klucza szyfrującego i deszyfrującego. Lokalizacje tych błędów zostały zinterpretowane jako fingerprinty i posłużyły do identyfikacji piratów. Niestety, metoda przedstawiona w [37, 96, 97] szyfruje i wysyła obraz do każdego użytkownika niezależnie, za pomocą wielu transmisji unicast, zatem nie posiada wysokiej wydajności konstrukcji i dystrybucji jakimi charakteryzują się metody JFD. Ponadto, fingerprinty osadzone są w dziedzinie przestrzennej, co sprawia, że metoda ta jest wrażliwa na kompresję, filtrowanie, szum, itp. Wnioski wyciągnięte przy badaniach właściwości tej metody pozwoliły na opracowanie całkowicie nowej metody JFD, opartej na rotacji kwaternionów [98], wolnej od powyższych wad.

W proponowanej metodzie JFD używana jest kwaternionowa reprezentacja obrazów kolorowych, w której trzy kanały kolorów RGB stanowią 3-wymiarową przestrzeń, a każdy element kolorowego obrazu jest punktem w tej przestrzeni. Proponowana metoda wykorzystuje szyfr bazujący na obustronnym iloczynie kwaternionów (ang. *quaternion sandwich product*) oraz trybie wiązania kwaternionów zaszyfrowanych CQC (ang. *Cipher-Quaternion Chaining*). Geometryczną interpretacją szyfrowania jest rotacja i przesunięcie punktów w 3-wymiarowej przestrzeni kolorów. Strona dystrybucyjna realizuje szyfrowanie selektywne za pomocą klucza szyfrującego i wysyła zaszyfrowane dane za pomocą transmisji multicast do wszystkich odbiorców. Każda strona odbiorcza realizuje deszyfrowanie za pomocą unikatowego klucza deszyfrującego, w którym zawarta jest informacja o fingerprincie danego odbiorcy. Podczas deszyfrowania, w obrazie pojawiają się niezauważalne zmiany, które są unikatowym fingerprintem danego odbiorcy. Zarówno szyfrowanie, jak i osadzanie fingerprintów jest realizowane w dziedzinie DCT, co z połączeniu z szyfrowaniem selektywnym zwiększa odporność osadzonych fingerprintów na ataki przetwarzania sygnałów. Ponadto, zgodnie z ideą JFD, proponowana metoda charakteryzuje się wydajnością konstrukcji i dystrybucji, dzięki jednokrotnemu szyfrowaniu, osadzaniu fingerprintów po stronie odbiorczej i dystrybucji multicast.

W kolejnych podrozdziałach przedstawiono opis proponowanej metody JFD, opartej na rotacji kwaternionów, w trybie wiązania kwaternionów zaszyfrowanych, składający się z następujących cześci: konstrukcja kluczy, szyfrowanie obrazu, realizowane po stronie nadawczej, łączny fingerprinting i deszyfracja, realizowany po stronie odbiorczej, oraz identyfikacja piratów, realizowana przez stronę nadawczą lub zaufaną trzecią stronę. Ponadto, przeprowadzono symulacyjne badania niezauważalności osadzonych fingerprintów, odporności na szum, odporności na kompresję, odporności na ataki zmowy oraz odporności na ataki zmowy połączone z kompresją. Wyniki przeprowadzonych badań zamieszczono w rozdziale 9.

7.1. Konstrukcja kluczy

Klucz szyfrujący $\mathbf{e}$ jest następującym wektorem o rozmiarze R:

$$\mathbf{e} = \begin{bmatrix} e_1 & e_2 & \cdots & e_R \end{bmatrix}, \tag{7.1}$$

gdzie $\mathbf{e}$ jest wektorem klucza szyfrującego, e_r jest kwaternionem o składowych skalarnych będących losowymi liczbami całkowitymi, $r = 1,2,...,R$. Podczas badań metody, wykorzystano kwaterniony e_r, których składowe przyjmowały wartości ze zbioru $\{1,2,3,4\}$ losowane zgodnie z rozkładem równomiernym. Aczkolwiek, można przyjąć inny zbiór wartości, np. składowe tych kwaternionów mogą przyjmować wartości binarne $\{0,1\}$. Klucz szyfrujący $\mathbf{e}$ jest znany jedynie stronie nadawczej.

Fingerprint *u*-tego użytkownika o długości M zostaje zapisany w postaci $M/4$ kwaternionów, które następnie zostają połączone w wektor $\mathbf{f}^{(u)}$ o rozmiarze R:

$$\mathbf{f}^{(u)} = \begin{bmatrix} f_1^{(u)} & f_2^{(u)} & \cdots & f_R^{(u)} \end{bmatrix}, \tag{7.2}$$

gdzie $\mathbf{f}^{(u)}$ jest wektorem fingerprintu *u*-tego użytkownika, $f_r^{(u)}$ jest kwaternionem o składowych skalarnych, będących częścią ciągu fingeprintu *u*-tego użytkownika, $r = 1,2,...,R$, $u = 1,2,...,U$, natomiast U jest liczbą wszystkich użytkowników. Jeżeli długość ciągu stanowiącego fingerprint jest mniejsza niż $4 \cdot R$, to wektor $\mathbf{f}^{(u)}$ musi zostać uzupełniony do rozmiaru R dodatkowymi kwaternionami o zerowych składowych, które powinny być równomiernie rozproszone w całym wektorze. Podczas badań metody, w charakterze fingerprintów wykorzystano bipolarne ciągi Golda o długości 4096 zapisane w postaci 1024 kwaternionów.

Klucz deszyfrujący $\mathbf{d}^{(u)}$ *u*-tego użytkownika jest wektorem o wymiarze R i jest zależny od klucza szyfrującego oraz fingerprintu *u*-tego użytkownika w następujący sposób:

$$\mathbf{d}^{(u)} = \mathbf{e} + \alpha \cdot \mathbf{f}^{(u)}, \tag{7.3}$$

gdzie $\mathbf{d}^{(u)}$ jest wektorem klucza deszyfrującego *u*-tego użytkownika, $\mathbf{e}$ jest wektorem klucza szyfrującego, $\mathbf{f}^{(u)}$ jest wektorem fingerprintu *u*-tego użytkownika, natomiast α jest współczynnikiem siły osadzania fingerprintu. Współczynnik siły osadzania fingerprintu α reprezentuje kompromis między niezauważalnością i odpornością osadzonych fingerprintów. Zmniejszanie wartości współczynnika α spowoduje, że fingerprint będzie mniej widoczny w odszyfrowanym obrazie. Natomiast, zwiększanie wartości współczynnika α spowoduje, że fingerprint będzie bardziej odporny na próby usunięcia. Wartość współczynnika α powinna być wybrana w taki sposób, aby możliwe było osiągnięcie możliwie największej odporności fingerprintów, przy jednoczesnym zachowaniu ich niezauważalności. Podczas badań, wartość współczynnika α została wybrana eksperymentalnie. Wygenerowane klucze deszyfrujące $\mathbf{d}^{(u)}$ należy dostarczyć poszczególnym użytkownikom w bezpieczny sposób.

7.2. Szyfrowanie

Przed rozpoczęciem szyfrowania, obraz kolorowy musi zostać wstępnie przetworzony w celu utworzenia wektora $\mathbf{x}$, który jest nazywany wektorem kwaternionów obrazu i zawiera współczynniki DCT wybrane do szyfrowania selektywnego. Najpierw, obraz kolorowy zostaje rozłożony na trzy podobrazy składowe zgodnie z modelem przestrzeni kolorów RGB, czyli na podobraz zawierający składową koloru czerwonego, zielonego i niebieskiego. Każda ze składowych zostaje podzielona na bloki o rozmiarze 8×8 pikseli. Następnie, każdy blok zostaje niezależnie poddany dwuwymiarowej dyskretnej transformacji kosinusowej DCT i przekształcony do postaci bloku 8×8 współczynników DCT. Szyfrowanie, a w rezultacie również osadzanie fingerprintów, jest realizowane we wszystkich trzech składowych kolorów: czerwonej, zielonej i niebieskiej.

Szyfrowanie wszystkich współczynników najlepiej ukryje wszystkie właściwości obrazu. Jednakże, zgodnie z koncepcją szyfrowania selektywnego, przedstawioną w rozdziale 4.2., zalecane jest zaszyfrowanie jedynie perceptualnie istotnych współczynników DCT. W rezultacie, złożoność obliczeniowa i opóźnienie związane z szyfrowaniem zostaną zmniejszone, a jednocześnie estymacja niezaszyfrowanego obrazu o wysokiej jakości nie będzie możliwa. Należy zwrócić uwagę, że fingerprinty użytkowników będą osadzone tylko w zaszyfrowanych współczynnikach. Wówczas, następnym krokiem jest wybranie *N* współczynników DCT ze składowej każdego koloru, odpowiadających perceptualnie istotnym elementom obrazu. Perceptualnie istotne współczynniki, odpowiadające widmowym składowym niskoczęstotliwościowym, znajdują się w lewym górnym rogu każdego bloku DCT. Współczynniki wysokiego rzędu, które znajdują się w prawym dolnym rogu bloku, są wrażliwe na kompresję, filtrowanie dolnopasmowe, skalowanie, rozmycie, itp., zatem nie mogą być wybrane do szyfrowania, ponieważ spowodowałoby to, że osadzone fingerprinty byłyby wrażliwe na powyższe operacje przetwarzania obrazu. Należy zwrócić uwagę, iż współczynnik DC, znajdujący się w pierwszym rzędzie i pierwszej kolumnie każdego bloku, również nie może być wybrany do osadzania fingerprintu, ponieważ jego, nawet niewielka, zmiana w znaczący sposób pogarsza jakość perceptualną obrazu. W celu rozproszenia fingerprintu na całej powierzchni obrazu, współczynniki wybrane do szyfrowania powinny pochodzić z możliwie jak największej liczby bloków DCT. Dla uproszczenia, bez utraty ogólności, niech liczba wszystkich bloków $B = N$. Wówczas, należy wybrać po jednym współczynniku z każdego bloku DCT, na przykład współczynniki w pierwszym rzędzie i drugiej kolumnie każdego bloku.

Wybrane współczynniki DCT ze składowej koloru czerwonego, zielonego i niebieskiego tworzą odpowiednio następujące wektory $\mathbf{x}^{(\mathrm{red})}$, $\mathbf{x}^{(\mathrm{green})}$, $\mathbf{x}^{(\mathrm{blue})}$:

$$\mathbf{x}^{(\mathrm{red})} = \left[x_1^{(\mathrm{red})} \quad x_2^{(\mathrm{red})} \quad \cdots \quad x_N^{(\mathrm{red})} \right], \tag{7.4}$$

$$\mathbf{x}^{(\mathrm{green})} = \left[x_1^{(\mathrm{green})} \quad x_2^{(\mathrm{green})} \quad \cdots \quad x_N^{(\mathrm{green})} \right], \tag{7.5}$$

$$\mathbf{x}^{(\mathrm{blue})} = \left[x_1^{(\mathrm{blue})} \quad x_2^{(\mathrm{blue})} \quad \cdots \quad x_N^{(\mathrm{blue})} \right], \tag{7.6}$$

gdzie $x_n^{(\text{red})}$, $x_n^{(\text{green})}$, $x_n^{(\text{blue})}$ są wybranymi współczynnikami DCT z odpowiednio czerwonej, zielonej i niebieskiej składowej obrazu, $n = 1, 2, ..., N$, natomiast N jest liczbą współczynników DCT, wybranych do szyfrowania selektywnego, z każdej składowej obrazu.

Należy zwrócić uwagę, iż scenariusz szyfrowania jednego współczynnika na blok jest jedynie przykładem i istnieje możliwość, aby szyfrować więcej albo mniej współczynników. Jeżeli $N < B$, to należy wybrać po jednym współczynniku z co $\lfloor B/N \rfloor$-tego bloku DCT, gdzie $\lfloor a \rfloor$ oznacza zaokrąglenie liczby *a* w dół do liczby całkowitej. Natomiast, jeżeli $N > B$, to wówczas należy wybrać po $\lceil N/B \rceil$ współczynników z każdego bloku DCT w kolejności zigzag (ang. *zigzag order*), gdzie $\lceil a \rceil$ oznacza zaokrąglenie liczby *a* w górę do liczby całkowitej. Współczynniki mogą być również wybierane w sposób losowy, ale algorytm wyboru współczynników musi być znany stronie odbiorczej.

W kolejnym kroku utworzony zostaje wektor kwaternionów obrazu $\mathbf{x}$, który zawiera współczynniki DCT wybrane do szyfrowania selektywnego z każdej składowej koloru i przedstawione w postaci kwaternionów:

$$x_n = 0 + x_n^{(\text{red})} i + x_n^{(\text{green})} j + x_n^{(\text{blue})} k, \tag{7.7}$$

$$\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_N \end{bmatrix}, \tag{7.8}$$

gdzie $\mathbf{x}$ jest wektorem kwaternionów obrazu, x_n jest *n*-tym kwaternionem obrazu reprezentującym punkt w 3-wymiarowej przestrzeni kolorów, którego współrzędne odpowiadają wartościom *n*-tych współczynników DCT wybranych ze składowej czerwonej, zielonej i niebieskiej, $n = 1, 2, ..., N$, natomiast N jest liczbą wybranych współczyników z każdej składowej.

Wektor $\mathbf{x}$ jest szyfrowany za pomocą rotacji kwaternionowej z przesunięciem:

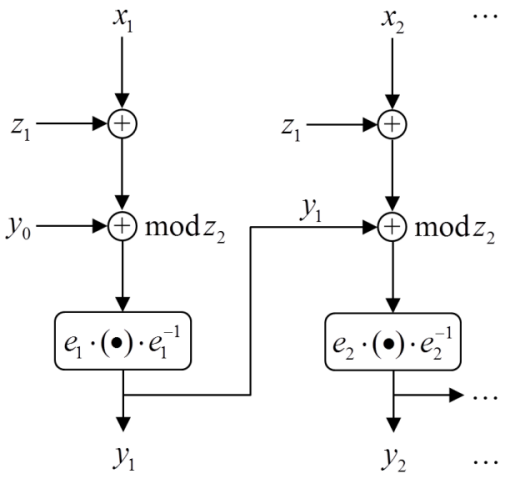
$$y_n = e_l \cdot [(x_n + z_1 + y_{n-1}) \bmod z_2] \cdot e_l^{-1}, \tag{7.9}$$

$$l = [(n-1) \bmod R] + 1, \tag{7.10}$$

$$\mathbf{y} = \begin{bmatrix} y_1 & y_2 & \cdots & y_N \end{bmatrix}, \tag{7.11}$$

gdzie $\mathbf{y}$ jest wektorem zaszyfrowanych kwaternionów obrazu, y_n jest *n*-tym zaszyfrowanym kwaternionem obrazu, y_{n-1} jest poprzednim zaszyfrowanym kwaternionem, y_0 jest kwaternionem inicjalizującym, x_n jest *n*-tym kwaternionem obrazu, e_l jest *l*-tym kwaternionem klucza szyfrującego, $n = 1, 2, ..., N$, N jest liczbą wybranych współczynników z każdej składowej, R jest rozmiarem wektora klucza szyfrującego, z_1 jest kwaternionem przesunięcia, z_2 jest bazą arytmetyki modularnej. Kwaternion inicjujący y_0 jest kwaternionem o zerowej składowej rzeczywistej i niezerowych składowych urojonych, np. $y_0 = 0 + 10i + 10j + 10k$.

Kwaternion przesunięcia z_1 jest stałym kwaternionem wprowadzonym w celu zwiększenia wartości wszystkich współczynników DCT do dodatnich wartości w celu uproszczenia realizacji sumy modulo z_2. W trakcie badań przyjęto: $z_1 = 0 + 1000i + 1000j + 1000k$ oraz $z_2 = 2000$. Modularne sumowanie kwaternionu obrazu x_n oraz poprzedniego zaszyfrowanego kwaternionu y_{n-1} zostało wprowadzone dla realizacji trybu wiązania kwaternionów zaszyfrowanych, który, przez analogię do oznaczenia trybu wiązania bloków zaszyfrowanych CBC (ang. *Cipher-Block Chaining*) dla szyfrów blokowych, oznaczono CQC (ang. *Cipher-Quaternion Chaining*). Zasada CQC została przedstawiona na rysunku 7.1.



Rys. 7.1. Szyfrowanie w trybie wiązania kwaternionów zaszyfrowanych CQC w metodzie JFD opartej na rotacji kwaternionów, opracowanie własne.

Zaszyfrowane kwaterniony y_n, będące elementami wektora $\mathbf{y}$, reprezentują punkty w 3-wymiarowej przestrzeni kolorów, które w wyniku szyfrowania zostały obrócone i przesunięte zgodnie z kluczem szyfrującym. Współrzędne tych punktów stanowią zaszyfrowane współczynniki DCT dla składowej koloru czerwonego, zielonego i niebieskiego:

$$y_n = 0 + y_n^{(\text{red})} i + y_n^{(\text{green})} j + y_n^{(\text{blue})} k, \tag{7.12}$$

gdzie $y_n^{(\text{red})}$, $y_n^{(\text{green})}$ $y_n^{(\text{blue})}$ są zaszyfrowanymi wybranymi współczynnikami DCT dla odpowiednio czerwonej, zielonej i niebieskiej składowej obrazu, $n = 1, 2, ..., N$, natomiast N jest liczbą współczyników wybranych z każdej składowej. Następnie, zaszyfrowane współczynniki DCT $y_n^{(\text{red})}$, $y_n^{(\text{green})}$ $y_n^{(\text{blue})}$ należy umieścić w ich lokalizacjach w blokach DCT każdej składowej, z których zostały wcześniej wybrane do szyfrowania selektywnego.

W ostatnim kroku, następuje przetwarzanie końcowe wszystkich bloków DCT, tzn.: bloki zostają poddane kwantyzacji DCT, ustawione w kolejności zigzag, oraz poddane kompresji bezstratnej za pomocą kodowania metodą RLE (ang. *Run-Length Encoding*) i metodą Huffmana. Należy zwrócić uwagę, iż szyfrowanie jest przeprowadzane jednokrotnie, niezależnie od liczby użytkowników systemu. Zaszyfrowane dane są wysyłane do wszystkich użytkowników za pomocą jednej transmisji rozgłoszeniowej typu mulicast.

7.3. Deszyfrowanie

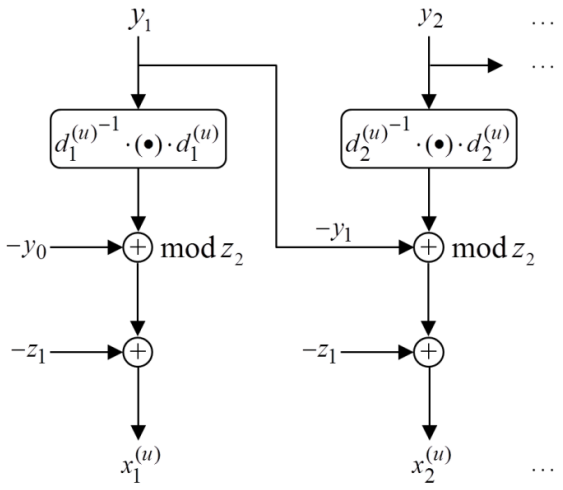
W celu odszyfrowania danych po stronie odbiorczej, najpierw należy przeprowadzić przetwarzanie wstępne, które zawiera dekodowanie RLE i Huffmana, odwrócenie kolejności zigzag oraz dekwantyzację DCT odebranego obrazu. Następnie, strona odbiorcza musi odtworzyć wektor $\mathbf{y}$ zaszyfrowanych kwaternionów y_n na podstawie zaszyfrowanych współczynników DCT składowej czerwonej $y_n^{(\text{red})}$, zielonej $y_n^{(\text{green})}$ i niebieskiej $y_n^{(\text{blue})}$, $n = 1,2,...,N$, a N jest liczbą współczyników wybranych z każdej składowej. Po rekonstrukcji wektora $\mathbf{y}$, łączny fingerprinting i deszyfracja dla *u*-tego użytkownika realizowane następująco:

$$x_n^{(u)} = [(d_l^{(u)^{-1}} \cdot y_n \cdot d_l^{(u)} - y_{n-1}) \bmod z_2] - z_1, \tag{7.13}$$

$$l = [(n-1) \bmod R] + 1, \tag{7.14}$$

$$\mathbf{x}^{(u)} = \left[x_1^{(u)} \quad x_2^{(u)} \quad \cdots \quad x_N^{(u)}\right], \tag{7.15}$$

gdzie $\mathbf{x}^{(u)}$ jest wektorem odszyfrowanych i oznakowanych kwaternionów obrazu, $x_n^{(u)}$ jest *n*-tym odszyfrowanym kwaternionem obrazu z fingerprintem *u*-tego użytkownika, y_n jest *n*-tym zaszyfrowanym kwaternionem obrazu, y_{n-1} jest poprzednim zaszyfrowanym kwaternionem obrazu, y_0 jest kwaternionem inicjalizującym, $d_l^{(u)}$ jest *l*-tym kwaternionem klucza deszyfrującego *u*-tego użytkownika, $n = 1,2,...,N$, N jest liczbą wybranych współczynników z każdej składowej, R jest rozmiarem wektora klucza deszyfrującego, z_1 jest kwaternionem przesunięcia, z_2 jest bazą arytmetyki modularnej. Deszyfrowanie, w trybie wiązania zaszyfrowanych kwaternionów, zostało przedstawione na rysunku 7.2.



Rys. 7.2. Deszyfrowanie w trybie wiązania kwaternionów zaszyfrowanych CQC w metodzie JFD opartej na rotacji kwaternionów, opracowanie własne.

Odszyfrowane kwaterniony $x_n^{(u)}$, będące elementami wektora $\mathbf{x}^{(u)}$, reprezentują punkty w 3-wymiarowej przestrzeni kolorów, które w wyniku deszyfrowania zostały obrócone i przesunięte zgodnie z kluczem deszyfrującym. Różnice między kluczem szyfrującym $\mathbf{e}$ oraz kluczem deszyfrujacym *u*-tego użytkownika $\mathbf{d}^{(u)}$, opisane wzorem (7.3), powodują osadzenie fingerprintu podczas deszyfrowania (7.13). Wówczas, wektor $\mathbf{x}^{(u)}$ różni się od wektora $\mathbf{x}$ o osadzony fingerprint $\mathbf{f}^{(u)}$. Współrzędne kwaternionów $x_n^{(u)}$ stanowią odszyfrowane współczynniki DCT dla składowej koloru czerwonego, zielonego i niebieskiego, które należy umieścić w ich lokalizacjach w blokach DCT, z których zostały wcześniej wybrane. W ostatnim kroku, następuje przetwarzanie końcowe wszystkich bloków DCT odszyfrowanej składowej koloru czerwonego, zielonego i niebieskiego z osadzonym fingerprintem *u*-tego użytkownika. Każdy blok zostaje niezależnie poddany dwuwymiarowej odwrotnej dyskretnej transformacji kosinusowej IDCT i przekształcony do postaci bloku 8×8 pikseli. Następnie, wszystkie bloki o rozmiarze 8×8 pikseli zostają połączone w celu odtworzenia trzech podobrazów składowych zgodnie z modelem przestrzeni kolorów RGB. Ostatecznie, podobrazy składowe RGB zostają połączone w odszyfrowany obraz kolorowy z osadzonym fingerprintem *u*-tego użytkownika. Pomimo, że fingerprint został osadzony tylko w wybranych współczynnikach DCT, to po powrocie do domeny przestrzennej, tj. wartości pikseli obrazu, będzie on rozproszony we wszystkich pikselach każdego bloku, który zawiera odszyfrowany współczynnik DCT.

7.4. Identyfikacja piratów

Wydobycie fingerprintu z przechwyconej pirackiej kopii i identyfikacja piratów odpowiedzialnych za wygenerowanie tej kopii jest realizowana przez stronę dystrybucyjną lub zaufaną trzecią stronę. Zakłada się, że strona dystrybucyjna lub zaufana trzecia strona ma dostęp do oryginalnych danych oraz fingerprintów użytkowników. Wówczas, możliwe jest zastosowanie detekcji koherentnej.

W pierwszym kroku, należy przeprowadzić wydobycie fingerprintu obecnego w pirackiej kopii obrazu. W tym celu, strona dystrybucyjna musi odtworzyć wektor $\mathbf{x}$ z obrazu oryginalnego, zgodnie z opisem przedstawionym wzorami od (7.4) do (7.8) w podrozdziale 7.2. Następnie, należy przeprowadzić analogiczne operacje dla pirackiego obrazu w celu otrzymania wektora $\mathbf{x}_\mathrm{P}$ zawierającego kwaterniony pirackiego obrazu. Fingerprint obecny w pirackiej kopii jest wydobywany w następujący sposób:

$$\mathbf{f}_\mathrm{P} = \frac{1}{\alpha} \cdot (\mathbf{x}_\mathrm{P} - \mathbf{x}), \tag{7.16}$$

gdzie $\mathbf{f}_\mathrm{P}$ jest wektorem kwaternionów fingerprintu wydobytego z pirackiej kopii, $\mathbf{x}_\mathrm{P}$ jest wektorem kwaternionów obrazu pirackiego z fingerprintem pirata, $\mathbf{x}$ jest wektorem kwaternionów obrazu oryginalnego bez fingerprintu, natomiast α jest współczynnikiem siły osadzania fingerprintu.

W drugim kroku, należy obliczyć właściwą postać fingerprintów wszystkich użytkowników. Należy zwrócić uwagę, iż właściwy fingerprint *u*-tego użytkownika $\mathbf{f}_\mathrm{emb}^{(u)}$ osadzony

w wektorze $\mathbf{x}^{(u)}$ nie jest w postaci $\mathbf{f}^{(u)}$, jak to ma miejsce w klasycznym modelu opisanym wzorem (2.2). Można wykazać, że osadzony fingerprint $\mathbf{f}_{\text{emb}}^{(u)}$ zależy w silnie nieliniowy sposób od wektora obrazu $\mathbf{x}^{(u)}$, fingerprintu $\mathbf{f}^{(u)}$ oraz klucza szyfrującego $\mathbf{e}$. Wyprowadzenie tej zależności zamieszczono w załączniku A. W celu identyfikacji piratów, strona dystrybucyjna lub zaufana trzecia strona, mając dostęp do kluczy deszyfrujących, oblicza wektory kwaternionów obrazu oznakowanego $\mathbf{x}^{(u)}$ wszystkich użytkowników, a następnie oblicza wektory kwaternionów osadzonych fingerprintów $\mathbf{f}_{\text{emb}}^{(u)}$ poszczególnych użytkowników w sposób analogiczny do równości (7.16):

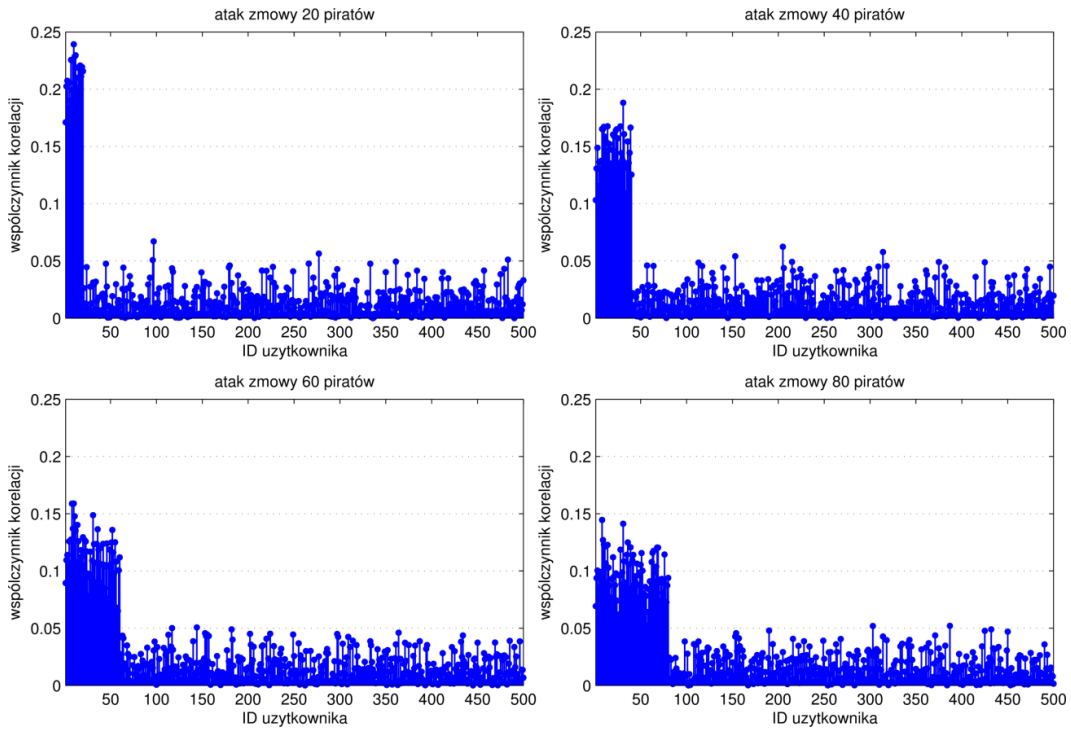
$$\mathbf{f}_{\text{emb}}^{(u)} = \frac{1}{\alpha} \cdot (\mathbf{x}^{(u)} - \mathbf{x}) . \tag{7.17}$$

gdzie $\mathbf{f}_{\text{emb}}^{(u)}$ jest wektorem kwaternionów fingerprintu wydobytego z oznakowanej kopii *u*-tego użytkownika, $\mathbf{x}^{(u)}$ jest wektorem kwaternionów obrazu z fingerprintem *u*-tego użytkownika, $\mathbf{x}$ jest wektorem kwaternionów obrazu oryginalnego bez fingerprintu, α jest współczynnikiem siły osadzania fingerprintu, $u = 1,2,...,U$, natomiast U jest liczbą użytkowników.

W ostatnim kroku realizowana jest analiza korelacyjna. Obliczany jest zbiór U współczynników korelacji skrośnej $\rho^{(u)}$, $u = 1,2,...U$, między fingerprintem wydobytym z piracktej kopii $\mathbf{f}_{\text{P}}$ oraz fingerprintem każdego użytkownika $\mathbf{f}_{\text{emb}}^{(u)}$:

$$\rho^{(u)} = \left| \frac{\sum_{n=1}^{N}\sum_{a=1}^{3} (\text{im}(f_{n\text{P}}, a) - \overline{\mathbf{f}_{\text{P}}})(\text{im}(f_{n_{\text{emb}}}^{(u)}, a) - \overline{\mathbf{f}_{\text{emb}}^{(u)}})}{\sqrt{\left(\sum_{n=1}^{N}\sum_{a=1}^{3} (\text{im}(f_{n\text{P}}, a) - \overline{\mathbf{f}_{\text{P}}})^2\right)\left(\sum_{n=1}^{N}\sum_{a=1}^{3} (\text{im}(f_{n_{\text{emb}}}^{(u)}, a) - \overline{\mathbf{f}_{\text{emb}}^{(u)}})^2\right)}} \right|, \tag{7.18}$$

gdzie $\rho^{(u)}$ jest wartością bezwzględną współczynnika korelacji wektorów kwaternionów $\mathbf{f}_{\text{P}}$ i $\mathbf{f}_{\text{emb}}^{(u)}$, $\text{im}(q, a)$ jest funkcją zwracającą współczynnik skalarny przy *a*-tej składowej urojonej kwaternionu *q*, $\overline{\mathbf{f}_{\text{P}}}$ jest wartością średnią współczynników skalarnych przy wszystkich składowych urojonych we wszystkich kwaternionach wektora $\mathbf{f}_{\text{P}}$, natomiast $\overline{\mathbf{f}_{\text{emb}}^{(u)}}$ jest wartością średnią współczynników skalarnych przy wszystkich składowych urojonych we wszystkich kwaternionach wektora $\mathbf{f}_{\text{emb}}^{(u)}$. Przykłady obliczonych współczynników korelacji, przy różnej liczbie zmawiających się piratów, przedstawiono na rysunku 7.3. Następnie, obliczone współczynniki korelacji $\rho^{(u)}$ są porównywane z ustalonym progiem detekcji t_{D}. Jeżeli współczynnik korelacji $\rho^{(u)}$ dla fingerprintu *u*-tego użytkownika jest większy od progu detekcji t_{D}, to *u*-ty użytkownik jest zidentyfikowany jako pirat.



Rys. 7.3. Przykłady wartości współczynników korelacji między fingerprintem wydobytym z pirackiej kopii oraz fingerprintem każdego użytkownika dla metody JFD opartej na rotacji kwaternionów (piraci mają początkowe wartości identyfikatorów), opracowanie własne.

Na rysunku 7.3. przedstawiono przykładowe wartości współczynników korelacji dla fingerprintu wydobytego z pirackiej kopii oraz fingerprintów poszczególnych użytkowników. Przedstawiono cztery przypadki, w których piracka kopia została utworzona przez 20, 40, 60 lub 80 piratów. Dla ułatwienia analizy wykresów, w każdym ataku zmowy, zmawiający się piraci mieli początkowe wartości identyfikatorów. Można zaobserwować, że wartości unormowanych współczynników korelacji dla fingerprintów piratów są znacznie większe niż wartości współczynników dla fingerprintów niewinnych użytkowników, co umożliwia identyfikację piratów.

7.5. Podsumowanie

W tym rozdziale przedstawiono nową metodę łącznego fingerprintingu i deszyfracji opartą na rotacji kwaternionów, realizującą szyfrowanie w trybie wiązania kwaternionów zaszyfrowanych CQC. Proponowana metoda wykorzystuje kwaternionową reprezentację elementów obrazu w 3-wymiarowej przestrzeni kolorów oraz szyfrowanie kwaternionowe składowej koloru czerwonego, zielonego i niebieskiego.

Metoda charakteryzuje się wysoką wydajnością konstrukcji, gdyż obraz jest szyfrowany jednokrotnie, niezależnie od liczby użytkowników, a osadzanie fingerprintu jest realizowane po stronie odbiorczej. Dzięki temu, wymagania na zasoby obliczeniowe strony nadawczej nie rosną wraz z liczbą klientów. Ponadto, metoda charakteryzuje się również wysoką wydajnością dystrybucji, ponieważ wystarczy tylko jedna transmisja multicas w celu dostarczenia oznakowanych kopii do wszystkich użytkowników. Oznacza to, że przepustowość wymagana do obsługi wszystkich klientów, nie rośnie wraz z ich liczbą.

W celu zmniejszenia czasu szyfrowania i deszyfrowania, w proponowanej metodzie wykorzystano koncepcję szyfrowania selektywnego. Obraz nie jest szyfrowany w całości, ale szyfrowane są jedynie wybrane, istotne perceptualnie, współczynniki DCT. Stopień zaszyfrowania obrazu może być regulowany poprzez zmianę liczby wybranych współczynników DCT. W celu osiągnięcia wysokiej odporności na ataki przez przetwarzanie sygnału obrazu, np. dodawanie szumu lub kompresję, wszystkie operacje są wykonywane w dziedzinie dyskretnej transformaty kosinusowej. Ponadto, po powrotnej konwersji do dziedziny przestrzennej, osadzony fingerprint jest rozproszony na całej powierzchni obrazu, dzięki czemu nie może zostać wycięty w całości z obrazu, np. w ataku kadrującym.

W powyższym opisie metody, do określenia 3-wymiarowej przestrzeni kolorów wykorzystywany jest model przestrzeni kolorów RGB. Oczywiście, istnieje możliwość wykorzystania innych modeli, np. YCbCr, aczkolwiek, przestrzeń kolorów RGB jest bardziej użyteczna z następujących względów. Współczynniki DCT, na odpowiadających sobie pozycjach w blokach, przyjmują wartości z takiego samego zakresu w każdym kolorze składowym modelu RGB. Wówczas, możliwe jest wykorzystanie takiego samego współczynnika siły osadzania fingerprintów α w każdej składowej, co upraszcza algorytm. W przypadku wykorzystania modelu YCbCr, współczynniki DCT przyjmują znacznie większe wartości w składowej luminancji niż w składowych chrominancji, zatem konieczne byłoby takie zmodyfikowanie metody, aby umożliwiała ona zastosowanie różnych współczynników α dla luminancji i chrominancji. Ponadto, fingerprinty osadzone równomiernie w każdym kolorze składowym RGB są mniej wrażliwe na kompresję (w standardzie JPEG dla obrazów lub w standardzie MPEG dla wideo), niż w przypadku użycia przestrzeni YCbCr. Jest to spowodowane tym, że kompresja bardzo silnie wpływa na składowe chrominancji, powodując znaczne zniekształcenia części fingerprintu. W przypadku wykorzystania modelu YCbCr, bardzo silnemu zniekształceniu ulega aż 2/3 osadzonego fingerprintu. Natomiast, w przypadku wykorzystania modelu RGB, zniekształcenia wynikające z kompresji są równomiernie rozłożone w całym fingerprincie, co w rezultacie powoduje, że w ogólności fingerprint jest mniej uszkodzony, a detekcja piratów skuteczniejsza.

8. METODA JFD OPARTA NA SERII ROTACJI KWATERNIONÓW

W tym rozdziale została zaproponowana nowa metoda łącznego fingerprintingu i deszyfracji oparta na serii rotacji kwaternionów. Inspiracją do powstania tej metody była metoda oparta na rotacji kwaternionów w trybie CQC, przedstawiona w rozdziale 7. Zadaniem projektowym, przyświecającym opracowywaniu nowej metody, było sprawdzenie, jak zwiększenie złożoności algorytmu szyfrowania, poprzez zwiększenie liczby rotacji, wpłynie na niezauważalność oraz odporność osadzonych fingerprintów.

W proponowanej metodzie JFD używana jest kwaternionowa reprezentacja obrazów kolorowych, w której trzy kanały kolorów RGB stanowią 3-wymiarową przestrzeń, a każdy element kolorowego obrazu jest punktem w tej przestrzeni. Geometryczną interpretacją szyfrowania jest przesunięcie i seria rotacji punktów w 3-wymiarowej przestrzeni kolorów. Strona dystrybucyjna realizuje szyfrowanie selektywne za pomocą klucza szyfrującego i wysyła zaszyfrowane dane, za pomocą transmisji multicast, do wszystkich odbiorców. Każdy odbiorca realizuje deszyfrowanie za pomocą unikatowego klucza deszyfrującego, który wygenerowano na podstawie klucza szyfrującego oraz fingerprintu danego odbiorcy. Podczas deszyfrowania, w obrazie osadzany jest niezauważalny fingerprint odbiorcy. Zarówno szyfrowanie, jak i osadzanie fingerprintów jest realizowane w dziedzinie DCT, co zwiększa odporność osadzonych fingerprintów na ataki za pomocą metod przetwarzania sygnałów. Ponadto, zgodnie z ideą JFD, proponowana metoda charakteryzuje się wydajnością konstrukcji i dystrybucji, dzięki jednokrotnemu szyfrowaniu, osadzaniu fingerprintów po stronie odbiorczej i dystrybucji multicast.

W kolejnych podrozdziałach przedstawiono opis proponowanej metody JFD, opartej na serii rotacji kwaternionów, składający się z następujących części: konstrukcja kluczy, szyfrowanie obrazu, realizowane po stronie nadawczej, łączny fingerprinting i deszyfracja, realizowane po stronie odbiorczej, oraz identyfikacja piratów, realizowana przez stronę nadawczą lub zaufaną trzecią stronę. Ponadto, przeprowadzono badania symulacyjne niezauważalności osadzonych fingerprintów, odporności na szum, odporności na kompresję, odporności na ataki zmowy oraz odporności na ataki zmowy połączone z kompresją. Wyniki przeprowadzonych badań zamieszczono w rozdziale 9.

8.1. Konstrukcja kluczy

Klucz szyfrujący $\mathbf{E}$ jest macierzą kwaternionów o rozmiarach $R_1 \times R_2$:

$$\mathbf{E} = \begin{bmatrix} e_{1,1} & e_{1,2} & \cdots & e_{1,R_2} \\ e_{2,1} & e_{2,2} & \cdots & e_{2,R_2} \\ \vdots & \vdots & \ddots & \vdots \\ e_{R_1,1} & e_{R_1,2} & \cdots & e_{R_1,R_2} \end{bmatrix}, \tag{8.1}$$

gdzie $\mathbf{E}$ jest macierzą klucza szyfrującego, złożoną z kwaternionów e_{r_1,r_2} o składowych skalarnych, będących losowymi liczbami całkowitymi, $r_1 = 1,2,...,R_1$, $r_2 = 1,2,...,R_2$. Kwaternio-

ny e_{r_1,r_2} muszą być kwaternionami znormalizowanymi. Podczas badań metody, wykorzystano kwaterniony e_{r_1,r_2}, których wartości składowe były losowane ze zbioru $\{1,2,3,4\}$, zgodnie z rozkładem równomiernym, a następnie, tak otrzymane kwaterniony, były normowane. Klucz szyfrujący $\mathbf{E}$ jest znany jedynie stronie nadawczej.

Fingerprint *u*-tego użytkownika o długości M, gdzie $M \le 4 \cdot R_1 \cdot R_2$, zostaje zapisany w postaci $M/4$ kwaternionów w macierzy $\mathbf{F}^{(u)}$ o rozmiarze $R_1 \times R_2$:

$$\mathbf{F}^{(u)} = \begin{bmatrix} f_{1,1}^{(u)} & f_{1,2}^{(u)} & \cdots & f_{1,R_2}^{(u)} \\ f_{2,1}^{(u)} & f_{2,2}^{(u)} & \cdots & f_{2,R_2}^{(u)} \\ \vdots & \vdots & \ddots & \vdots \\ f_{R_1,1}^{(u)} & f_{R_1,2}^{(u)} & \cdots & f_{R_1,R_2}^{(u)} \end{bmatrix}, \tag{8.2}$$

gdzie $\mathbf{F}^{(u)}$ jest macierzą fingerprintu *u*-tego użytkownika, $f_{r_1,r_2}^{(u)}$ jest kwaternionem o składowych skalarnych, będących częścią ciągu fingerprintu *u*-tego użytkownika, $r_1 = 1,2,...,R_1$, $r_2 = 1,2,...,R_2$, $u = 1,2,...,U$, natomiast U jest liczbą wszystkich użytkowników. Jeżeli długość ciągu stanowiącego fingerprint jest mniejsza niż $4 \cdot R_1 \cdot R_2$, to macierz $\mathbf{F}^{(u)}$ musi zostać uzupełniona dodatkowymi kwaternionami o zerowych składowych, które powinny być równomiernie rozproszone w całej macierzy. Podczas badań metody, w charakterze fingerprintów wykorzystano bipolarne ciągi Golda o długości 4096 zapisane w postaci 1024 kwaternionów.

Klucz deszyfrujący $\mathbf{D}^{(u)}$ *u*-tego użytkownika jest macierzą o wymiarze $R_1 \times R_2$:

$$\mathbf{D}^{(u)} = \begin{bmatrix} d_{1,1}^{(u)} & d_{1,2}^{(u)} & \cdots & d_{1,R_2}^{(u)} \\ d_{2,1}^{(u)} & d_{2,2}^{(u)} & \cdots & d_{2,R_2}^{(u)} \\ \vdots & \vdots & \ddots & \vdots \\ d_{R_1,1}^{(u)} & d_{R_1,2}^{(u)} & \cdots & d_{R_1,R_2}^{(u)} \end{bmatrix}, \tag{8.3}$$

$$d_{r_1,r_2}^{(u)} = \frac{e_{r_1,r_2} \cdot (1 + \alpha \cdot f_{r_1,r_2}^{(u)})}{\left\| e_{r_1,r_2} \cdot (1 + \alpha \cdot f_{r_1,r_2}^{(u)}) \right\|}, \tag{8.4}$$

gdzie $\mathbf{D}^{(u)}$ jest macierzą klucza deszyfrującego *u*-tego użytkownika, $d_{r_1,r_2}^{(u)}$ jest kwaternionem klucza deszyfrującego *u*-tego użytkownika, e_{r_1,r_2} jest kwaternionem klucza szyfrującego, $f_{r_1,r_2}^{(u)}$ jest kwaternionem fingerprintu *u*-tego użytkownika $r_1 = 1,2,...,R_1$, $r_2 = 1,2,...,R_2$, $u = 1,2,...,U$, U jest liczbą wszystkich użytkowników, natomiast α jest współczynnikiem siły osadzania fingerprintu. Kwaterniony $d_{r_1,r_2}^{(u)}$ są kwaternionami znormalizowanymi. Współczynnik siły osadzania fingerprintu α reprezentuje kompromis między niezauważalnością i odpornością osadzonych fingerprintów. Zmniejszanie wartości współczynnika α spowoduje, że fingerprint będzie mniej widoczny w odszyfrowanym obrazie. Natomiast, zwiększanie wartości współ-

czynnika α spowoduje, że fingerprint będzie bardziej odporny na próby usunięcia. Wartość współczynnika α powinna być wybrana w taki sposób, aby możliwe było osiągnięcie możliwie największej odporności fingerprintów, przy jednoczesnym zachowaniu ich niezauważalności. Podczas badań, wartość współczynnika α została wybrana eksperymentalnie. Wygenerowane klucze deszyfrujące $\mathbf{D}^{(u)}$ należy dostarczyć poszczególnym użytkownikom w bezpieczny sposób.

8.2. Szyfrowanie

Przed rozpoczęciem szyfrowania, obraz kolorowy musi zostać wstępnie przetworzony w celu konstrukcji macierzy $\mathbf{X}$, która jest nazywana macierzą kwaternionów obrazu i zawiera współczynniki DCT, wybrane do szyfrowania selektywnego. Najpierw, obraz kolorowy zostaje rozłożony na trzy podobrazy składowe, zgodnie z modelem przestrzeni kolorów RGB, czyli na podobraz zawierający składową koloru czerwonego, zielonego i niebieskiego. Każda ze składowych zostaje podzielona na bloki o rozmiarze 8×8 pikseli. Następnie, każdy blok zostaje niezależnie poddany dwuwymiarowej dyskretnej transformacji kosinusowej DCT i przekształcony do postaci bloku 8×8 współczynników DCT. Szyfrowanie, a w rezultacie również osadzanie fingerprintów, jest realizowane jednocześnie we wszystkich trzech składowych kolorów: czerwonej, zielonej i niebieskiej.

Szyfrowanie wszystkich współczynników skutecznie ukryje wszystkie właściwości obrazu, ale w celu zmniejszenia opóźnienia związanego z szyfrowaniem, przy jednoczesnym uniemożliwieniu estymacji obrazu oryginalnego o wysokiej jakości, zalecane jest szyfrowanie selektywne, czyli szyfrowanie jedynie perceptualnie istotnych współczynników DCT. Fingerprinty zostaną osadzone tylko w zaszyfrowanych współczynnikach. Do szyfrowania selektywnego, konieczne jest wybranie $N_1 \times N_2$ współczynników DCT ze składowej każdego koloru, odpowiadających perceptualnie istotnym elementom obrazu, gdzie $N_1 \times N_2$ będzie rozmiarem macierzy obrazu $\mathbf{X}$. Perceptualnie istotne współczynniki, odpowiadające widmowym składowym niskoczęstotliwościowym, znajdują się w lewym górnym rogu każdego bloku DCT. Współczynniki wysokiego rzędu, które znajdują się w prawym dolnym rogu bloku, są wrażliwe na kompresję, filtrowanie dolnopasmowe, skalowanie, rozmycie, itp., zatem nie mogą być wybrane do szyfrowania, ponieważ spowodowałoby to, że osadzone fingerprinty byłyby wrażliwe na powyższe operacje przetwarzania obrazu. Należy też zwrócić uwagę, iż współczynnik DC, znajdujący się w pierwszym rzędzie i pierwszej kolumnie każdego bloku, również nie może być wybrany do osadzania fingerprintu, ponieważ jego zmiana może silnie pogorszyć jakość perceptualną obrazu. W celu rozproszenia fingerprintu na całej powierzchni obrazu, współczynniki wybrane do szyfrowania powinny pochodzić z możliwie jak największej liczby bloków DCT. Dla uproszczenia, bez utraty ogólności, niech liczba wszystkich bloków $B = N_1 \cdot N_2$. Wówczas, należy wybrać po jednym współczynniku z każdego bloku DCT, na przykład współczynniki leżące w pierwszym rzędzie i drugiej kolumnie każdego bloku.

Wybrane współczynniki DCT ze składowej koloru czerwonego, zielonego i niebieskiego tworzą odpowiednio następujące macierze $\mathbf{X}^{(\mathrm{red})}$, $\mathbf{X}^{(\mathrm{green})}$, $\mathbf{X}^{(\mathrm{blue})}$:

$$\mathbf{X}^{(\mathrm{red})} = \begin{bmatrix} x_{1,1}^{(\mathrm{red})} & x_{1,2}^{(\mathrm{red})} & \cdots & x_{1,N_2}^{(\mathrm{red})} \\ x_{2,1}^{(\mathrm{red})} & x_{2,2}^{(\mathrm{red})} & \cdots & x_{2,N_2}^{(\mathrm{red})} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N_1,1}^{(\mathrm{red})} & x_{N_1,2}^{(\mathrm{red})} & \cdots & x_{N_1,N_2}^{(\mathrm{red})} \end{bmatrix}, \tag{8.5}$$

$$\mathbf{X}^{(\mathrm{green})} = \begin{bmatrix} x_{1,1}^{(\mathrm{green})} & x_{1,2}^{(\mathrm{green})} & \cdots & x_{1,N_2}^{(\mathrm{green})} \\ x_{2,1}^{(\mathrm{green})} & x_{2,2}^{(\mathrm{green})} & \cdots & x_{2,N_2}^{(\mathrm{green})} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N_1,1}^{(\mathrm{green})} & x_{N_1,2}^{(\mathrm{green})} & \cdots & x_{N_1,N_2}^{(\mathrm{green})} \end{bmatrix}, \tag{8.6}$$

$$\mathbf{X}^{(\mathrm{blue})} = \begin{bmatrix} x_{1,1}^{(\mathrm{blue})} & x_{1,2}^{(\mathrm{blue})} & \cdots & x_{1,N_2}^{(\mathrm{blue})} \\ x_{2,1}^{(\mathrm{blue})} & x_{2,2}^{(\mathrm{blue})} & \cdots & x_{2,N_2}^{(\mathrm{blue})} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N_1,1}^{(\mathrm{blue})} & x_{N_1,2}^{(\mathrm{blue})} & \cdots & x_{N_1,N_2}^{(\mathrm{blue})} \end{bmatrix}, \tag{8.7}$$

gdzie $x_{n_1,n_2}^{(\mathrm{red})}$, $x_{n_1,n_2}^{(\mathrm{green})}$, $x_{n_1,n_2}^{(\mathrm{blue})}$ są wybranymi współczynnikami DCT z odpowiednio czerwonej, zielonej i niebieskiej składowej obrazu, $n_1 = 1,2,...,N_1$, $n_2 = 1,2,...,N_2$ a $N_1 \cdot N_2$ jest liczbą współczynników DCT wybranych do szyfrowania selektywnego, z każdej składowej obrazu.

Stopień zaszyfrowania obrazu może być regulowany poprzez zmianę liczby wybranych współczynników. Należy wyraźnie podkreślić, iż opisywany scenariusz szyfrowania jednego współczynnika na blok jest jedynie przykładem i nic nie stoi na przeszkodzie, aby szyfrować więcej albo mniej współczynników. Jeżeli $N_1 \cdot N_2 < B$, to należy wybrać po jednym współczynniku z co $\lfloor B / N_1 \cdot N_2 \rfloor$-tego bloku DCT, gdzie $\lfloor a \rfloor$ oznacza zaokrąglenie liczby *a* w dół do najbliższej liczby całkowitej. Natomiast, jeżeli $N_1 \cdot N_2 > B$, to wówczas należy wybrać po $\lceil N_1 \cdot N_2 / B \rceil$ współczynników z każdego bloku DCT w kolejności zigzag (ang. *zigzag order*), gdzie $\lceil a \rceil$ oznacza zaokrąglenie liczby *a* w górę do najbliższej liczby całkowitej. Współczynniki mogą być również wybierane w sposób losowy, ale w takim przypadku należy zapewnić, aby algorytm wyboru współczynników był znany stronie odbiorczej.

W kolejnym kroku, utworzona zostaje macierz kwaternionów obrazu $\mathbf{X}$, która zawiera współczynniki DCT wybrane do szyfrowania selektywnego z każdej składowej koloru i które są przedstawione w postaci kwaternionów:

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,N_2} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,N_2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N_1,1} & x_{N_1,2} & \cdots & x_{N_1,N_2} \end{bmatrix}, \tag{8.9}$$

$$x_{n_1,n_2} = 0 + x_{n_1,n_2}^{(\mathrm{red})} i + x_{n_1,n_2}^{(\mathrm{green})} j + x_{n_1,n_2}^{(\mathrm{blue})} k, \tag{8.8}$$

gdzie $\mathbf{X}$ jest macierzą obrazu, x_{n_1,n_2} jest kwaternionem obrazu reprezentującym punkt w 3-wymiarowej przestrzeni kolorów, którego współrzędne (tzn. składowe urojone kwaternionu) są równe wartościom odpowiadających sobie współczynników DCT wybranych ze składowej czerwonej, zielonej i niebieskiej, $n_1 = 1,2,...,N_1$, $n_2 = 1,2,...,N_2$ a $N_1 \cdot N_2$ jest liczbą współczynników, wybranych do szyfrowania selektywnego, z każdej składowej obrazu.

Macierz $\mathbf{X}$ jest szyfrowana za pomocą serii rotacji kwaternionowych:

$$y_{n_1,n_2} = (\prod_{z=1}^{Z} e_{g_1(Z-z+1),g_2(Z-z+1)}) \cdot (x_{n_1,n_2} + \beta \cdot \mathrm{im}(e_{h_1,h_2})) \cdot (\prod_{z=1}^{Z} e_{g_1(z),g_2(z)}{}^*), \tag{8.10}$$

$$h_1 = [(n_1 - 1) \bmod R_1] + 1, \tag{8.11}$$

$$h_2 = [(n_2 - 1) \bmod R_2] + 1, \tag{8.12}$$

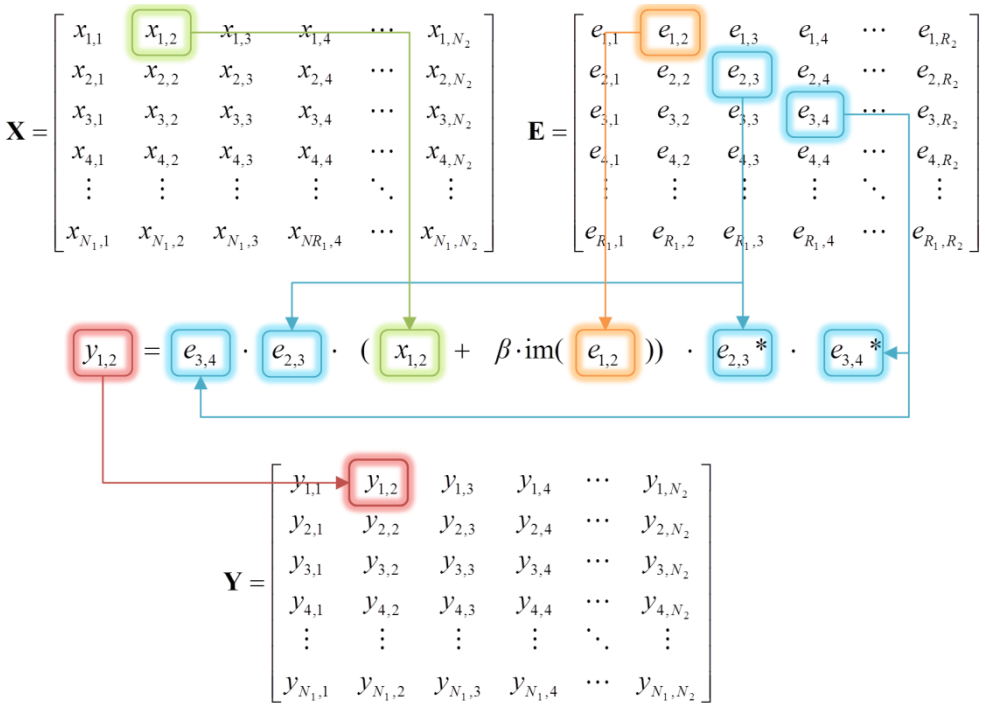
$$g_1(z) = [(n_1 - 1 + z) \bmod R_1] + 1, \tag{8.13}$$

$$g_2(z) = [(n_2 - 1 + z) \bmod R_2] + 1, \tag{8.14}$$

$$\mathbf{Y} = \begin{bmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,N_2} \\ y_{2,1} & y_{2,2} & \cdots & y_{2,N_2} \\ \vdots & \vdots & \ddots & \vdots \\ y_{N_1,1} & y_{N_1,2} & \cdots & y_{N_1,N_2} \end{bmatrix}, \tag{8.15}$$

gdzie $\mathbf{Y}$ jest macierzą zaszyfrowanych kwaternionów obrazu, y_{n_1,n_2} jest zaszyfrowanym kwaternionem obrazu, x_{n_1,n_2} jest kwaternionem obrazu, e_{h_1,h_2} są kwaternionami klucza szyfrującego, których współrzędne w macierzy $\mathbf{E}$ zostały wyjaśnione poniżej, $\mathrm{im}(\cdot)$ jest funkcją zwracającą część urojoną kwaternionu (czyli kwaternion o zerowej składowej rzeczywistej oraz niezmienionymi składowymi urojonymi), współczynnik β jest parametrem szyfrowania, Z jest długością serii rotacji kwaternionów, $z = 1, 2, ..., Z$, $n_1 = 1,2,...,N_1$, $n_2 = 1,2,...,N_2$ a $N_1 \cdot N_2$ jest liczbą wybranych współczynników z każdej składowej koloru obrazu. Współczynnik β wpływa na wartość przesunięcia kwaternionu szyfrowanego przed rotacją, a zatem w konsekwencji, wpływa na poziom zmian wizualnych w obrazie spowodowanych przez rotację. Zwiększenie wartości współczynnika β spowoduje zwiększenie odległości między kwaternionem przed rotacją, a początkiem układu współrzędnych, co w rezultacie spowoduje zwiększenie odległości między kwaternionem przed rotacją, a kwaternionem po rotacji. Zmiana współczynnika β może wymusić odpowiednie dopasowanie współczynnika siły osadzania α w celu zachowania niezauważalności osadzonych fingerprintów. Długość serii rotacji Z wpływa na złożoność szyfrowania. Zwiększanie długości serii Z spowoduje, że fingerprint będzie osadzany w deszyfrowanych kwaternionach w coraz bardziej złożony sposób. Niestety, zwiększanie długości serii rotacji Z wydłuża czasy szyfrowania i deszyfrowania.

Do zaszyfrowania jednego kwaternionu obrazu z macierzy $\mathbf{X}$ wykorzystywanych jest $Z+1$ kwaternionów klucza z macierzy $\mathbf{E}$. Najpierw, kwaternion obrazu x_{n_1,n_2}, z wiersza n_1 i kolumny n_2 macierzy $\mathbf{X}$, jest sumowany z częścią urojoną kwaternionu klucza e_{h_1,h_2} (przemnożoną przez współczynnik β), z wiersza h_1 i kolumny h_2 macierzy $\mathbf{E}$. Są to odpowiadające sobie pozycje w macierzach z uwzględnieniem ewentualnych różnic między rozmiarami tych macierzy, zgodnie z wzorami (8.11) i (8.12). Następnie, realizowana jest seria Z rotacji względem kwaternionów klucza $e_{g_1(z),g_2(z)}$ z rosnącymi indeksami obu współrzędnych, zgodnie z wzorami (8.13) i (8.14), czyli względem kolejnych kwaternionów leżących równolegle do przekątnej macierzy. Umiejscowienie kwaternionów klucza biorących udział w szyfrowaniu przykładowego kwaternionu obrazu $x_{1,2}$, przy długości serii $Z=2$, zostało zobrazowane na rysunku 8.1.



Rys. 8.1. Umiejscowienie kwaternionów klucza biorących udział w szyfrowaniu przykładowego kwaternionu obrazu $x_{1,2}$, przy długości serii $Z = 2$, opracowanie własne.

Należy zwrócić uwagę, że powyższy sposób wyboru kwaternionów z macierzy klucza szyfrującego $\mathbf{E}$, tzn. wybór kolejnych kwaternionów wzdłuż przekątnej macierzy, nie jest jedyną możliwością. Na przykład, można przyjąć prostszy wariant, w którym macierz klucza szyfrującego $\mathbf{E}$ oraz macierz obrazu $\mathbf{X}$ mają tylko jeden wiersz, czyli są wektorami jak w metodzie z rozdziału 7. Wówczas, kwaterniony klucza szyfrującego, wybierane do kolejnych rotacji w serii, są kolejnymi kwaternionami w macierzy $\mathbf{E}$. Takie podejście jest proste, ale posiada wadę polegającą na tym, że dwa sąsiednie kwaterniony obrazu zostaną zaszyfrowane z użyciem dwóch zbiorów kwaternionów klucza szyfrującego, różniących się tylko jednym kwaternionem. Natomiast, powyższy zaproponowany sposób, czyli wybór kwaternionów klucza szyfrującego wzdłuż przekątnej macierzy $\mathbf{E}$, zapewnia, że dwa sąsiednie kwaterniony, w tym samym wierszu lub kolumnie, będą zaszyfrowane przy pomocy dwóch rozłącznych

zbiorów kwaternionów klucza szyfrującego. Jeszcze inną regułą może być pseudolosowe wybieranie kwaternionów z macierzy klucza szyfrującego **E** do kolejnych rotacji w serii. Oczywiście, w takim przypadku generator liczb pseudolosowych oraz ziarno generatora muszą być ustalone ze stroną odbiorczą, aby deszyfracja była możliwa.

Zaszyfrowane kwaterniony y_{n_1,n_2}, będące elementami macierzy **Y**, reprezentują punkty w 3-wymiarowej przestrzeni kolorów, które w wyniku szyfrowania zostały przesunięte i obrócone zgodnie z kluczem szyfrującym. Współrzędne tych punktów stanowią zaszyfrowane współczynniki DCT dla składowej koloru czerwonego, zielonego i niebieskiego:

$$y_{n_1,n_2} = 0 + y_{n_1,n_2}^{(\text{red})} i + y_{n_1,n_2}^{(\text{green})} j + y_{n_1,n_2}^{(\text{blue})} k, \tag{8.16}$$

gdzie $y_{n_1,n_2}^{(\text{red})}$, $y_{n_1,n_2}^{(\text{green})}$, $y_{n_1,n_2}^{(\text{blue})}$ są zaszyfrowanymi wybranymi współczynnikami DCT dla odpowiednio czerwonej, zielonej i niebieskiej składowej obrazu, $n_1 = 1,2,...,N_1$, $n_2 = 1,2,...,N_2$ a $N_1 \cdot N_2$ jest liczbą współczynników wybranych z każdej składowej obrazu. Następnie, zaszyfrowane współczynniki $y_{n_1,n_2}^{(\text{red})}$, $y_{n_1,n_2}^{(\text{green})}$, $y_{n_1,n_2}^{(\text{blue})}$ należy umieścić w ich lokalizacjach w blokach każdej składowej, z których zostały wcześniej wybrane do szyfrowania selektywnego.

W ostatnim kroku, następuje przetwarzanie końcowe wszystkich bloków DCT, tzn.: bloki zostają poddane kwantyzacji DCT, ustawione w kolejności zigzag, oraz poddane kompresji bezstratnej za pomocą kodowania metodą RLE (ang. *Run-Length Encoding*) i metodą Huffmana. Należy zwrócić uwagę, iż szyfrowanie jest przeprowadzane jednokrotnie, niezależnie od liczby użytkowników systemu. Zaszyfrowane dane są wysyłane do wszystkich użytkowników za pomocą jednej transmisji rozgłoszeniowej typu mulicast.

8.3. Deszyfrowanie

W celu odszyfrowania danych po stronie odbiorczej, najpierw należy przeprowadzić przetwarzanie wstępne, które zawiera dekodowanie RLE i Huffmana, odwrócenie kolejności zigzag oraz dekwantyzację DCT odebranego obrazu. Następnie, strona odbiorcza musi odtworzyć macierz **Y** zaszyfrowanych kwaternionów y_{n_1,n_2} na podstawie zaszyfrowanych współczynników DCT składowej czerwonej $y_{n_1,n_2}^{(\text{red})}$, zielonej $y_{n_1,n_2}^{(\text{green})}$ i niebieskiej $y_{n_1,n_2}^{(\text{blue})}$, $n_1 = 1,2,...,N_1$, $n_2 = 1,2,...,N_2$ a $N_1 \cdot N_2$ jest liczbą współczynników wybranych z każdej składowej obrazu. Po rekonstrukcji macierzy **Y**, łączny fingerprinting i deszyfracja dla *u*-tego użytkownika realizowane następująco:

$$x_{n_1,n_2}^{(u)} = (\prod_{z=1}^{Z} d_{g_1(z),g_2(z)}^{(u)} *) \cdot y_{n_1,n_2} \cdot (\prod_{z=1}^{Z} d_{g_1(Z-z+1),g_2(Z-z+1)}^{(u)}) - \beta \cdot \text{im}(d_{h_1,h_2}^{(u)}), \tag{8.17}$$

$$h_1 = [(n_1 - 1) \bmod R_1] + 1, \tag{8.18}$$

$$h_2 = [(n_2 - 1) \bmod R_2] + 1, \tag{8.19}$$

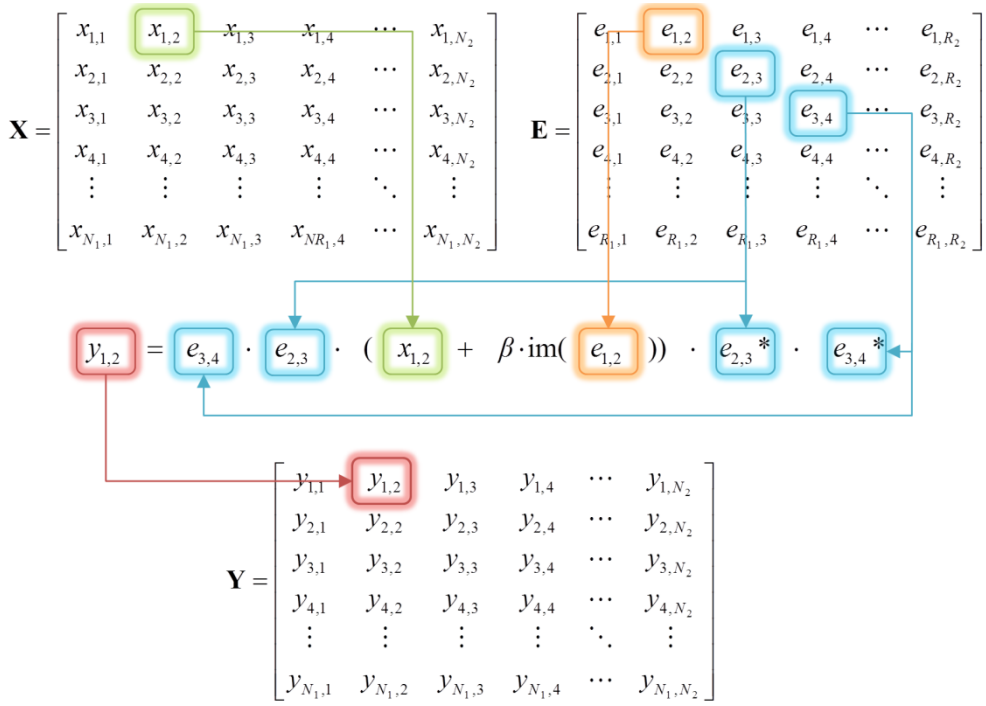
$$g_1(z) = [(n_1 - 1 + z) \bmod R_1] + 1, \tag{8.20}$$

$$g_2(z) = [(n_2 - 1 + z) \bmod R_2] + 1, \tag{8.21}$$

$$\mathbf{X}^{(u)} = \begin{bmatrix} x_{1,1}^{(u)} & x_{1,2}^{(u)} & \cdots & x_{1,N_2}^{(u)} \\ x_{2,1}^{(u)} & x_{2,2}^{(u)} & \cdots & x_{2,N_2}^{(u)} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N_1,1}^{(u)} & x_{N_1,2}^{(u)} & \cdots & x_{N_1,N_2}^{(u)} \end{bmatrix}, \tag{8.22}$$

gdzie $\mathbf{X}^{(u)}$ jest macierzą odszyfrowanych i oznakowanych kwaternionów obrazu, $x_{n_1,n_2}^{(u)}$ jest odszyfrowanym kwaternionem obrazu z fingerprintem *u*-tego użytkownika, y_{n_1,n_2} jest zaszyfrowanym kwaternionem obrazu, $d_{h_1,h_2}^{(u)}$ są kwaternionami klucza deszyfrującego *u*-tego użytkownika, których współrzędne w macierzy $\mathbf{D}^{(u)}$ są analogiczne jak w przypadku kwaternionów e_{h_1,h_2} w szyfrowaniu (wzór (8.10)), współczynnik β jest parametrem szyfrowania, Z jest długością serii rotacji kwaternionów, $z = 1,2,...,Z$, $n_1 = 1,2,...,N_1$, $n_2 = 1,2,...,N_2$ a $N_1 \cdot N_2$ jest liczbą współczynników, wybranych do szyfrowania selektywnego, z każdej składowej obrazu. Umiejscowienie kwaternionów klucza biorących udział w deszyfrowaniu przykładowego kwaternionu obrazu $y_{1,2}$, przy długości serii $Z = 2$, zostało pokazane na rysunku 8.2.

Odszyfrowane kwaterniony $x_{n_1,n_2}^{(u)}$, będące elementami macierzy $\mathbf{X}^{(u)}$, reprezentują punkty w 3-wymiarowej przestrzeni kolorów, które w wyniku deszyfrowania zostały obrócone i przesunięte zgodnie z kluczem deszyfrującym. Różnice między kluczem szyfrującym $\mathbf{E}$ oraz kluczem deszyfrującym *u*-tego użytkownika $\mathbf{D}^{(u)}$, opisane wzorem (8.4), powodują osadzenie fingerprintu podczas deszyfrowania (8.17). Wówczas, macierz $\mathbf{X}^{(u)}$ różni się od macierzy $\mathbf{X}$ o osadzony fingerprint $\mathbf{F}^{(u)}$. Współrzędne kwaternionów $x_{n_1,n_2}^{(u)}$ stanowią odszyfrowane współczynniki DCT dla składowej koloru czerwonego, zielonego i niebieskiego, które należy umieścić w ich lokalizacjach w blokach DCT, z których zostały wcześniej wybrane. W ostatnim kroku, następuje przetwarzanie końcowe wszystkich bloków DCT odszyfrowanych składowych kolorów z osadzonym fingerprintem *u*-tego użytkownika. Każdy blok zostaje niezależnie poddany dwuwymiarowej odwrotnej dyskretnej transformacji kosinusowej IDCT i przekształcony do postaci bloku 8×8 pikseli. Następnie, wszystkie bloki o rozmiarze 8×8 pikseli zostają połączone w celu odtworzenia trzech podobrazów składowych zgodnie z modelem przestrzeni kolorów RGB. Ostatecznie, podobrazy składowe RGB zostają połączone w odszyfrowany obraz kolorowy z osadzonym fingerprintem *u*-tego użytkownika. Pomimo, że fingerprint został osadzony tylko w wybranych współczynnikach DCT, to po powrocie do domeny przestrzennej, tj. wartości pikseli obrazu, będzie on rozproszony we wszystkich pikselach każdego bloku, który zawiera odszyfrowany współczynnik DCT.



Rys. 8.2. Umiejscowienie kwaternionów klucza biorących udział w deszyfrowaniu przykładowego kwaternionu obrazu $y_{1,2}$, przy długości serii $Z = 2$, opracowanie własne.

8.4. Identyfikacja piratów

Wydobycie fingerprintu z przechwyconej pirackiej kopii i identyfikacja piratów odpowiedzialnych za wygenerowanie tej kopii są realizowane przez stronę dystrybucyjną lub zaufaną trzecią stronę. Zakłada się, że strona dystrybucyjna lub zaufana trzecia strona ma dostęp do oryginalnych danych oraz fingerprintów użytkowników. Wówczas, możliwe jest zastosowanie detekcji koherentnej.

W pierwszym kroku, należy przeprowadzić ekstrakcję fingerprintu obecnego w pirackiej kopii obrazu. W tym celu, strona dystrybucyjna musi odtworzyć macierz $\mathbf{X}$ z obrazu oryginalnego, czyli wyodrębnić z obrazu składową czerwoną, zieloną i niebieską, każdą składową podzielić na bloki 8×8 pikseli, przekształcić je w bloki 8×8 współczynników DCT, wybrać odpowiednie, istotne perceptualnie współczynniki i zapisać je jako urojone składowe kwaternionów $x_{n_1,n_2}^{(u)}$ w macierzy $\mathbf{X}$. Następnie, należy przeprowadzić analogiczne operacje dla pirackiego obrazu w celu otrzymania macierzy $\mathbf{X}_\mathrm{P}$ zawierającej kwaterniony pirackiego obrazu. Fingerprint obecny w pirackiej kopii jest wydobywany w następujący sposób:

$$\mathbf{F}_\mathrm{P} = \frac{1}{\alpha} \cdot (\mathbf{X}_\mathrm{P} - \mathbf{X}), \tag{8.23}$$

gdzie $\mathbf{F}_\mathrm{P}$ jest macierzą kwaternionów fingerprintu wydobytego z pirackiej kopii, $\mathbf{X}_\mathrm{P}$ jest macierzą kwaternionów obrazu pirackiego z fingerprintem pirata, $\mathbf{X}$ jest macierzą kwaternionów obrazu oryginalnego bez fingerprintu, natomiast α jest współczynnikiem siły osadzania fingerprintu.

W drugim kroku, należy obliczyć właściwą postać fingerprintów wszystkich użytkowników. Należy zwrócić uwagę, iż właściwy fingerprint *u*-tego użytkownika $\mathbf{F}_{\text{emb}}^{(u)}$ osadzony w macierzy $\mathbf{X}^{(u)}$ nie jest w postaci $\mathbf{F}^{(u)}$, jak to ma miejsce w klasycznym modelu opisanym wzorem (2.2). Można wykazać, że osadzony fingerprint $\mathbf{F}_{\text{emb}}^{(u)}$ zależy w silnie nieliniowy sposób od macierzy obrazu $\mathbf{X}^{(u)}$, fingerprintu $\mathbf{F}^{(u)}$ oraz klucza szyfrującego $\mathbf{E}$. Wyprowadzenie tej zależności zamieszczono w załączniku B. W celu identyfikacji piratów, strona dystrybucyjna lub zaufana trzecia strona, mająca dostęp do wszystkich kluczy deszyfrujących, oblicza macierze kwaternionów obrazu oznakowanego $\mathbf{X}^{(u)}$ wszystkich użytkowników, a następnie oblicza macierze kwaternionów osadzonych fingerprintów $\mathbf{F}_{\text{emb}}^{(u)}$ poszczególnych użytkowników w sposób analogiczny do wzoru (8.23):

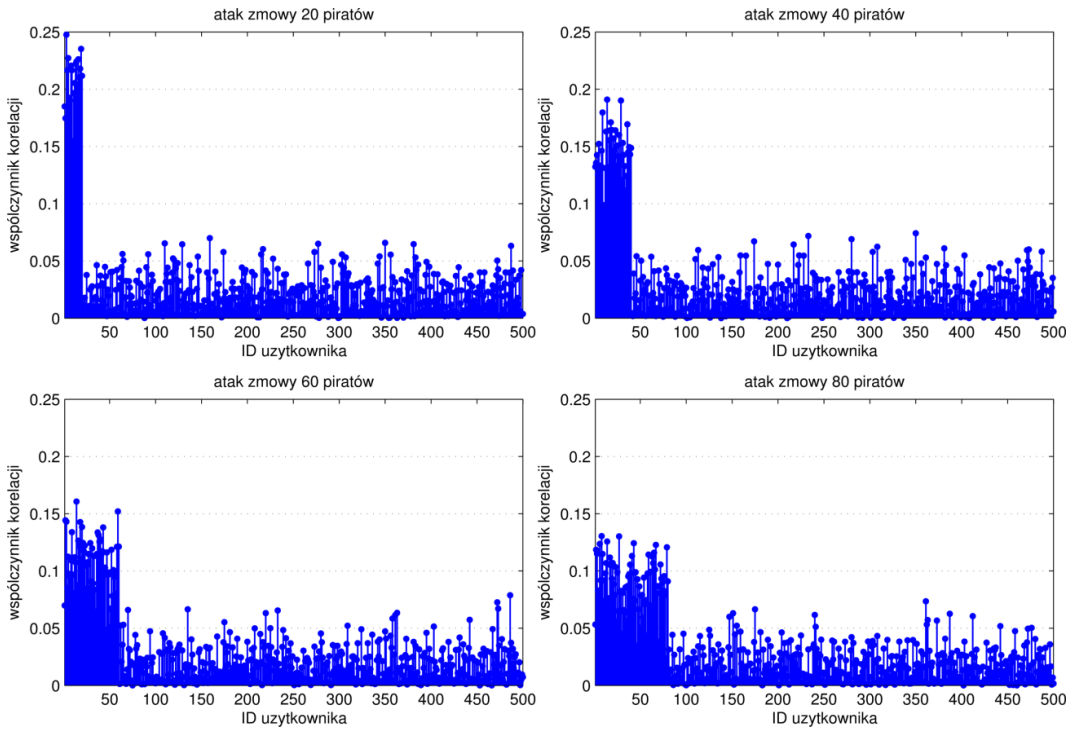
$$\mathbf{F}_{\text{emb}}^{(u)} = \frac{1}{\alpha} \cdot (\mathbf{X}^{(u)} - \mathbf{X}) . \tag{8.24}$$

gdzie $\mathbf{F}_{\text{emb}}^{(u)}$ jest macierzą kwaternionów fingerprintu wydobytego z oznakowanej kopii *u*-tego użytkownika, $\mathbf{X}^{(u)}$ jest macierzą kwaternionów obrazu z fingerprintem *u*-tego użytkownika, $\mathbf{X}$ jest macierzą kwaternionów obrazu oryginalnego bez fingerprintu, α jest współczynnikiem siły osadzania fingerprintu, $u = 1,2,...,U$, natomiast U jest liczbą użytkowników.

W ostatnim kroku, realizowana jest analiza korelacyjna. Obliczany jest zbiór U współczynników korelacji skrośnej $\rho^{(u)}$, $u = 1,2,...,U$, między fingerprintem wydobytym z piracki kopii $\mathbf{F}_{\text{P}}$ oraz fingerprintem każdego użytkownika $\mathbf{F}_{\text{emb}}^{(u)}$:

$$\rho^{(u)} = \left| \frac{\sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{a=1}^{3} (\text{im}(f_{r_1,r_2\,\text{P}}, a) - \overline{\mathbf{F}_{\text{P}}})(\text{im}(f_{r_1,r_2\,\text{emb}}^{(u)}, a) - \overline{\mathbf{F}_{\text{emb}}^{(u)}})}{\sqrt{\left(\sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{a=1}^{3} (\text{im}(f_{r_1,r_2\,\text{P}}, a) - \overline{\mathbf{F}_{\text{P}}})^2 \right) \left(\sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{a=1}^{3} (\text{im}(f_{r_1,r_2\,\text{emb}}^{(u)}, a) - \overline{\mathbf{F}_{\text{emb}}^{(u)}})^2 \right)}} \right| , \tag{8.25}$$

gdzie $\rho^{(u)}$ jest wartością bezwzględną współczynnika korelacji macierzy kwaternionów $\mathbf{F}_{\text{P}}$ i $\mathbf{F}_{\text{emb}}^{(u)}$, $\text{im}(q, a)$ jest funkcją zwracającą współczynnik skalarny przy *a*-tej składowej urojonej kwaternionu *q*, $\overline{\mathbf{F}_{\text{P}}}$ jest wartością średnią współczynników skalarnych przy wszystkich składowych urojonych we wszystkich kwaternionach macierzy $\mathbf{F}_{\text{P}}$, natomiast $\overline{\mathbf{F}_{\text{emb}}^{(u)}}$ jest wartością średnią współczynników skalarnych przy wszystkich składowych urojonych we wszystkich kwaternionach macierzy $\mathbf{F}_{\text{emb}}^{(u)}$. Przykłady obliczonych współczynników korelacji, przy różnej liczbie zmawiających się piratów, przedstawiono na rysunku 8.3. Następnie, obliczone współczynniki korelacji $\rho^{(u)}$ są porównywane z ustalonym progiem detekcji t_{D}. Jeżeli współczynnik korelacji $\rho^{(u)}$ dla fingerprintu *u*-tego użytkownika jest większy od progu detekcji t_{D}, to *u*-ty użytkownik zostaje zidentyfikowany jako pirat.



Rys. 8.3. Przykłady wartości współczynników korelacji między fingerprintem wydobytym z pirackiej kopii oraz fingerprintem każdego użytkownika dla metody JFD opartej na serii rotacji kwaternionów (piraci mają początkowe wartości identyfikatorów), opracowanie własne.

Na rysunku 8.3. przedstawiono przykładowe bezwzględne wartości współczynników korelacji między fingerprintem wydobytym z pirackiej kopii oraz fingerprintami poszczególnych użytkowników. Przedstawiono przypadki czterech ataków zmowy realizowanych przez 20, 40, 60 oraz 80 piratów. W każdym przypadku, zmawiającymi się piratami byli użytkownicy o najniższych wartościach identyfikatorów. Należy zwrócić uwagę, że wartości bezwzględne współczynników korelacji dla fingerprintów piratów są znacznie większe niż wartości współczynników dla fingerprintów niewinnych użytkowników, co umożliwia skuteczną identyfikację piratów.

8.5. Podsumowanie

W tym rozdziale przedstawiono nową metodę łącznego fingerprintingu i deszyfracji opartą na serii rotacji kwaternionów, dla obrazów kolorowych. Proponowana metoda wykorzystuje kwaternionową reprezentację elementów obrazu w 3-wymiarowej przestrzeni kolorów oraz szyfrowanie kwaternionowe składowej koloru czerwonego, zielonego i niebieskiego. W celu redukcji czasu szyfrowania i złożoności obliczeniowej algorytmu, w metodzie została wykorzystana koncepcja szyfrowania selektywnego, tzn. szyfrowane są jedynie wybrane, istotne perceptualnie, współczynniki DCT. Łączny fingerprinting i deszyfracji jest realizowany w dziedzinie dyskretnej transformaty kosinusowej, dzięki czemu metoda charakteryzuje się wysoką odpornością na dodawanie szumu oraz kompresję, ze względu na osadzanie fingerprintów w niskoczęstotliwościowych składowych widmowych obrazu, oraz na ataki kadrujące, ze względu na równomierne rozproszenie fingerprintów na całej powierzchni obrazu.

Podobnie jak metody przedstawione w rozdziale 5. i 7., proponowana metoda charakteryzuje się bardzo dużą wydajnością konstrukcji oraz bardzo dużą wydajnością dystrybucji. Wymagania na zasoby obliczeniowe strony nadawczej nie rosną wraz z liczbą klientów, ponieważ szyfrowanie jest realizowane jednokrotnie, a osadzanie fingerprintu jest realizowane po stronie odbiorczej. Natomiast, wymagania na przepustowość nie rosną wraz z liczbą odbiorców, gdyż wystarcza tylko jedna transmisja multicast w celu dostarczenia oznakowanych kopii do wszystkich użytkowników.

W powyższym opisie metody, do określenia 3-wymiarowej przestrzeni kolorów wykorzystywany jest model przestrzeni kolorów RGB. Podobnie jak w przypadku metody opartej na rotacji kwaternionów w trybie CQC, która została przedstawiona w rozdziale 7., możliwe jest wykorzystanie innych modeli kolorów, np. YCbCr, jednakże przestrzeń kolorów RGB jest bardziej użyteczna ze względów przedstawionych w podrozdziale 7.5.

9. WYNIKI BADAŃ

Celem pracy było zaprojektowanie skutecznej metody łącznego fingerprintingu i deszyfracji dla obrazów kolorowych. Jednocześnie założono, że fingerprinty muszą być osadzone w obrazie w sposób niezauważalny dla ludzkiego oka oraz, że projektowana metoda ma być odporna na próby usunięcia osadzanych fingerprintów. Przyjęto również, że głównym kryterium oceny skuteczności projektowanej metody będzie odporność na ataki zmowy. W celu zweryfikowania, czy założony cel został osiągnięty, przeprowadzono obszerne badania symulacyjne dla wszystkich trzech nowych metod JFD dla obrazów kolorowych, tzn. dla metody opartej na iloczynie macierzy, metody opartej na rotacji kwaternionów w trybie CQC, oraz metody opartej na serii rotacji kwaternionów. Przeprowadzone badania obejmowały badanie niezauważalności osadzonych fingerprintów, odporności na szum, odporności na kompresję, odporności na ataki zmowy, oraz odporności na ataki zmowy połączone z kompresją.

W tym rozdziale zamieszczono parametry zrealizowanych symulacji, osiągnięte wyniki badań, oraz porównanie nowych metod JFD i innych metod fingerprintingu przedstawionych w literaturze. W całym rozdziale 9., dla skrócenia wyrażeń w tekście, metodę JFD opartą na iloczynie macierzy opisaną w rozdziale 5. nazwano metodą A, metodę JFD opartą na rotacji kwaternionów z CQC opisaną w rozdziale 7. nazwano metodą B, a metodę JFD opartą na serii rotacji kwaternionów opisaną w rozdziale 8. nazwano metodą C.

9.1. Parametry symulacji

Podczas badań, zasymulowano następujący scenariusz usługi dystrybucji obrazów. Liczba wszystkich użytkowników usługi wynosiła $U = 500$. W roli dystrybuowanych obrazów użyto 10 kolorowych obrazów o rozmiarze 512×512 pikseli, które przedstawiają różne obiekty, osoby i miejsca, o różnym stopniu szczegółowości. Obrazy zostały zabezpieczone przed nielegalną redystrybucją poprzez osadzenie fingerprintów. Osadzanymi ciągami, stanowiącymi fingerprinty użytkowników, były bipolarne ciągi Golda o długości 4096, które w przypadku metody B i C zostały zapisane w 1024 kwaternionach. Dla metody A, macierz klucza szyfrującego miała rozmiar 64×64. Dla metody B, rozmiar wektora klucza szyfrującego wynosił $R = 1024$ kwaterniony. Dla metody C, rozmiar macierzy klucza szyfrującego wynosił 32×32 kwaterniony. W każdej metodzie, szyfrowano jeden współczynnik z każdego bloku DCT. Wartość współczynnika siły osadzania została ustalona eksperymentalnie, tak, aby zapewnić niezauważalność fingerprintów, i wynosiła $\alpha = 0{,}015$ dla metody A, $\alpha = 0{,}029$ dla metody B, i $\alpha = 0{,}0055$ przy $\beta = 1500$ dla metody C. Ponadto, podczas badania niezauważalności fingerprintów przeprowadzono jeszcze dodatkowe symulacje dla szyfrowania 16 współczynników z każdego bloku, z wartością współczynnika $\alpha = 0{,}007$ dla metody A, $\alpha = 0{,}007$ dla metody B, i $\alpha = 0{,}002$ przy $\beta = 1000$ dla metody C. Długość serii rotacji kwaternionowych w metodzie C była równa $Z = 2$. Dla każdej metody, symulacja dystrybucji każdego z 10 obrazów została powtórzona 5 razy, za każdym razem dla innego zestawu kluczy, co w rezultacie dało $S = 50$ powtórzeń symulacji eksperymentu. Założono, że w każdej symulacji część użytkowników jest piratami, którzy generują pirackie kopie obrazów w różnego rodzaju atakach. Opisy tych ataków i wyniki badania odporności na te ataki zostały

przedstawione w kolejnych podrozdziałach. Podczas badań odporności na ataki założono bardziej krytyczny przypadek, tj. szyfrowanie tylko jednego współczynnika na każdy blok DCT.

W celu analizy wyników, zdefiniowano następujące rezultaty detekcji i ich prawdopodobieństwa. Poprawnie pozytywna detekcja TP (ang. *True Positive*) oznacza, że pirat, który brał udział w ataku został poprawnie zidentyfikowany. Prawdopodobieństwo poprawnie pozytywnej detekcji P_{TP} dla danego ataku i progu detekcji obliczano w następujący sposób:

$$P_{TP} = \frac{I}{C \cdot S}, \tag{9.1}$$

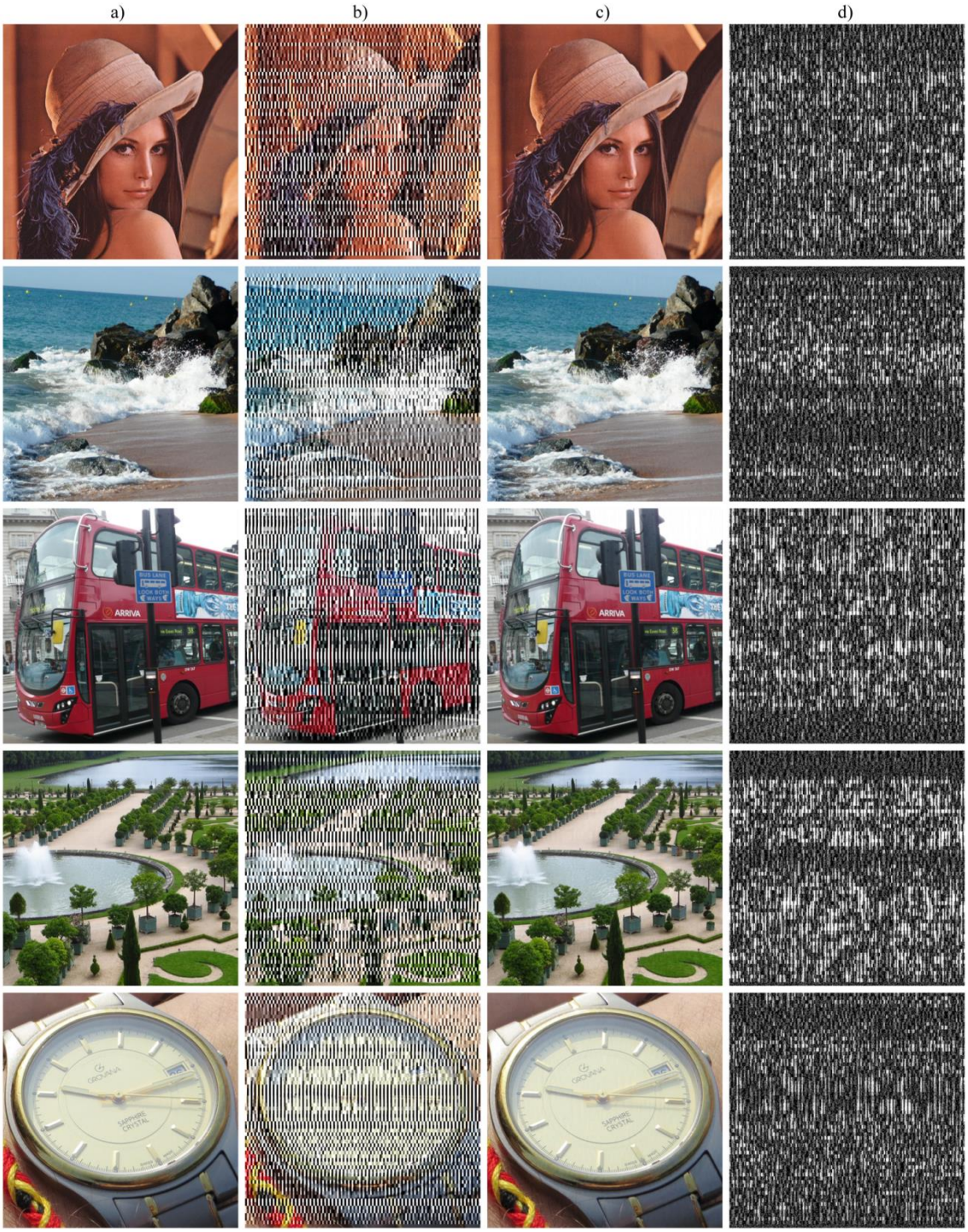
gdzie I jest liczbą poprawnie zidentyfikowanych piratów we wszystkich symulacjach dla danego progu detekcji, C jest liczbą wszystkich piratów w ataku, natomiast S jest liczbą zrealizowanych symulacji. Fałszywie pozytywna detekcja FP (ang. *False Positive*) oznacza, że uczciwy użytkownik został niesłusznie oskarżony. Prawdopodobieństwo fałszywie pozytywnej detekcji P_{FP} dla rozpatrywanego ataku i progu detekcji obliczano w następujący sposób:

$$P_{FP} = \frac{A}{(U - C) \cdot S}, \tag{9.2}$$

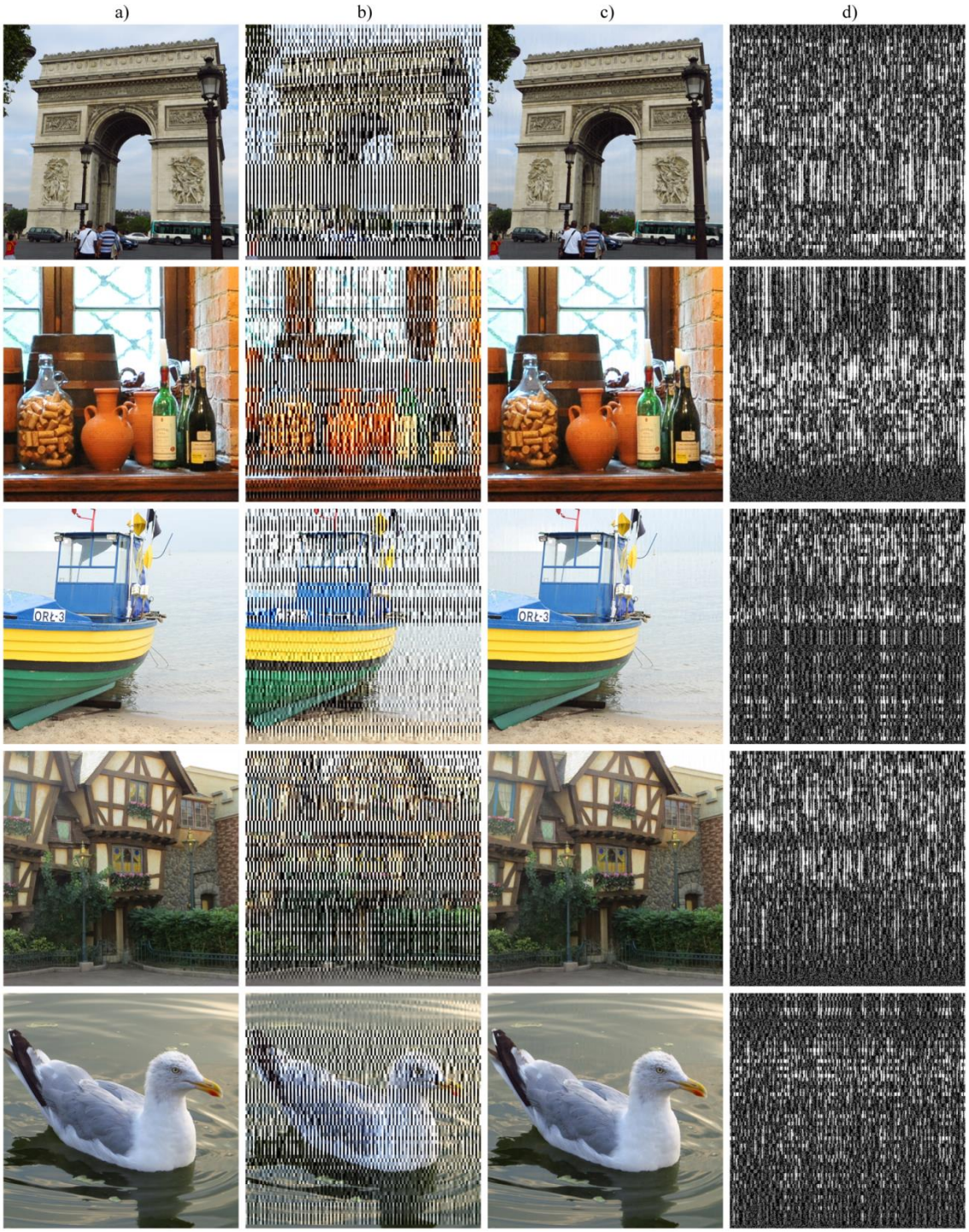
gdzie A jest liczbą niesłusznie oskarżonych użytkowników, C jest liczbą piratów w ataku, U jest liczbą wszystkich użytkowników, natomiast S jest liczbą zrealizowanych symulacji.

9.2. Niezauważalność osadzonych fingerprintów

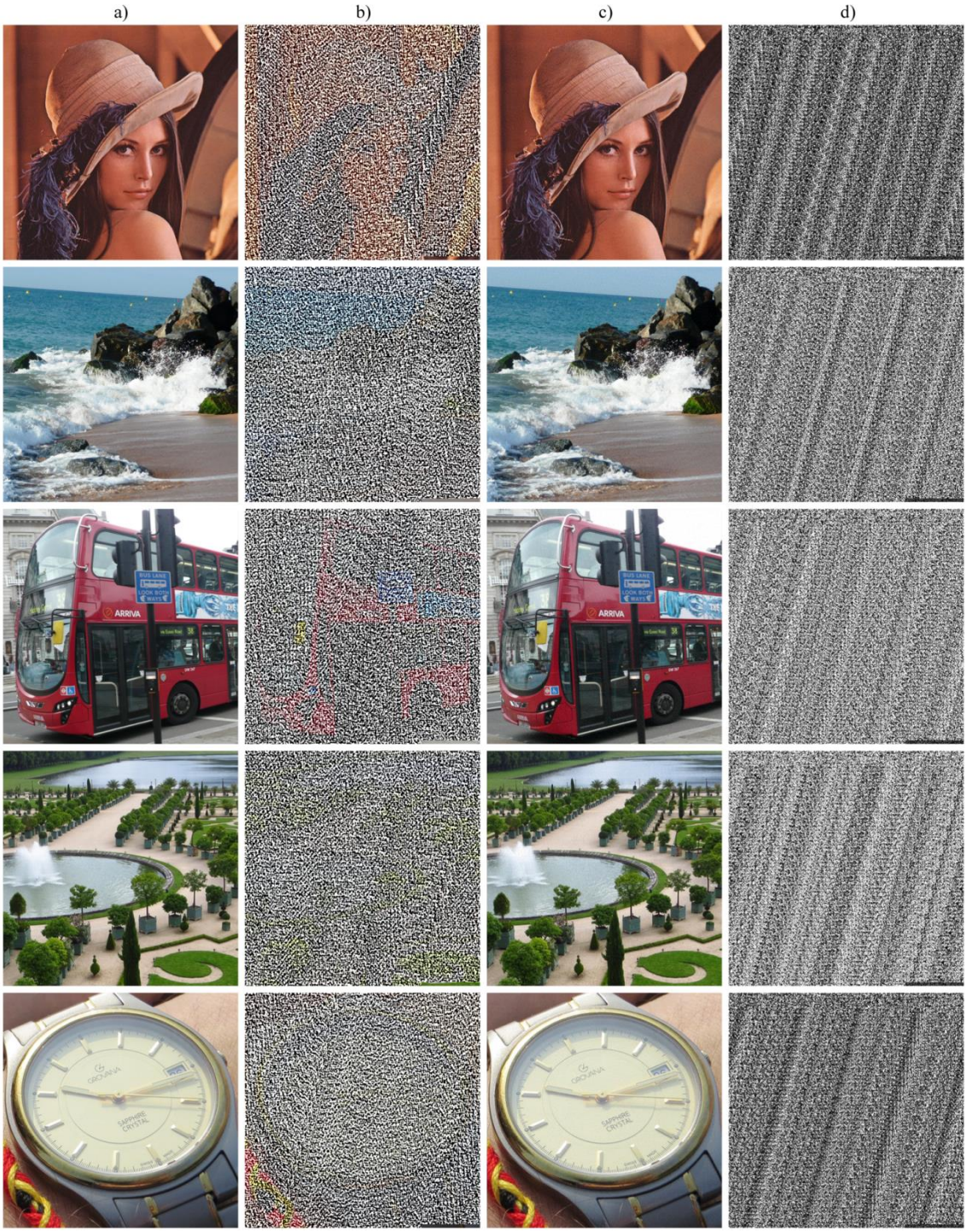
W przeprowadzonych badaniach, niezauważalność osadzonych fingerprintów została wstępnie oceniona w sposób subiektywny. Rysunki od 9.1. do 9.12. przedstawiają przykładowe obrazy w postaci: a) oryginalnej, b) zaszyfrowanej, c) odszyfrowanej z osadzonym fingerprintem oraz d) fingerprint wydobyty z obrazu odszyfrowanego Rysunki od 9.1. do 9.4. dotyczą metody A, rysunki od 9.5. do 9.8. dotyczą metody B, a rysunki od 9.9. do 9.12 dotyczą metody C. Rysunki 9.1., 9.2., 9.5., 9.6., 9.9., 9.10. przedstawiają przypadek szyfrowania tylko jednego współczynnika DCT na każdy blok, natomiast rysunki 9.3., 9.4., 9.7., 9.8., 9.11., 9.12. przedstawiają przypadek szyfrowania 16 współczynników na blok. Na rysunkach 9.1., 9.2., 9.5., 9.6., 9.9., 9.10., można zaobserwować, że przy szyfrowaniu jednego współczynnika na blok, część treści obrazu po zaszyfrowaniu jest widoczna, aczkolwiek komfortowe oglądanie takiego obrazu nie jest możliwe, co w niektórych zastosowaniach jest wystarczające. Jednakże, proponowana metoda daje możliwość skuteczniejszego ukrycia treści obrazu przez szyfrowanie większej liczby współczynników, czego efekt zaprezentowano na rysunkach 9.3., 9.4., 9.7., 9.8., 9.11., 9.12.



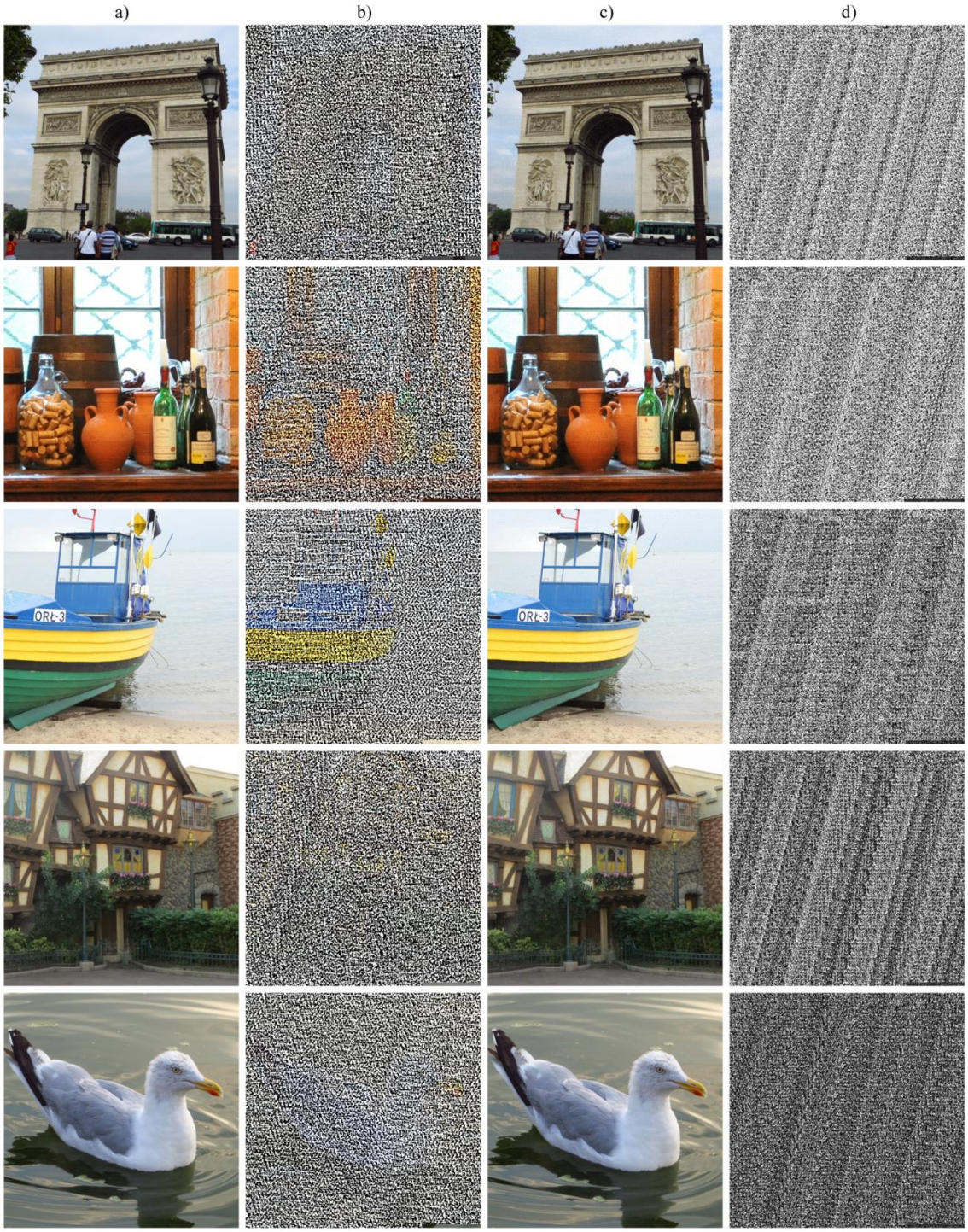
Rys. 9.1. Metoda A: przykładowe obrazy w postaci a) oryginalnej, b) zaszyfrowanej, c) odszyfrowanej z osadzonym fingerprintem oraz d) fingerprint wydobyty z obrazu odszyfrowanego, dla szyfrowania jednego współczynnika DCT w każdym bloku, opracowanie własne.



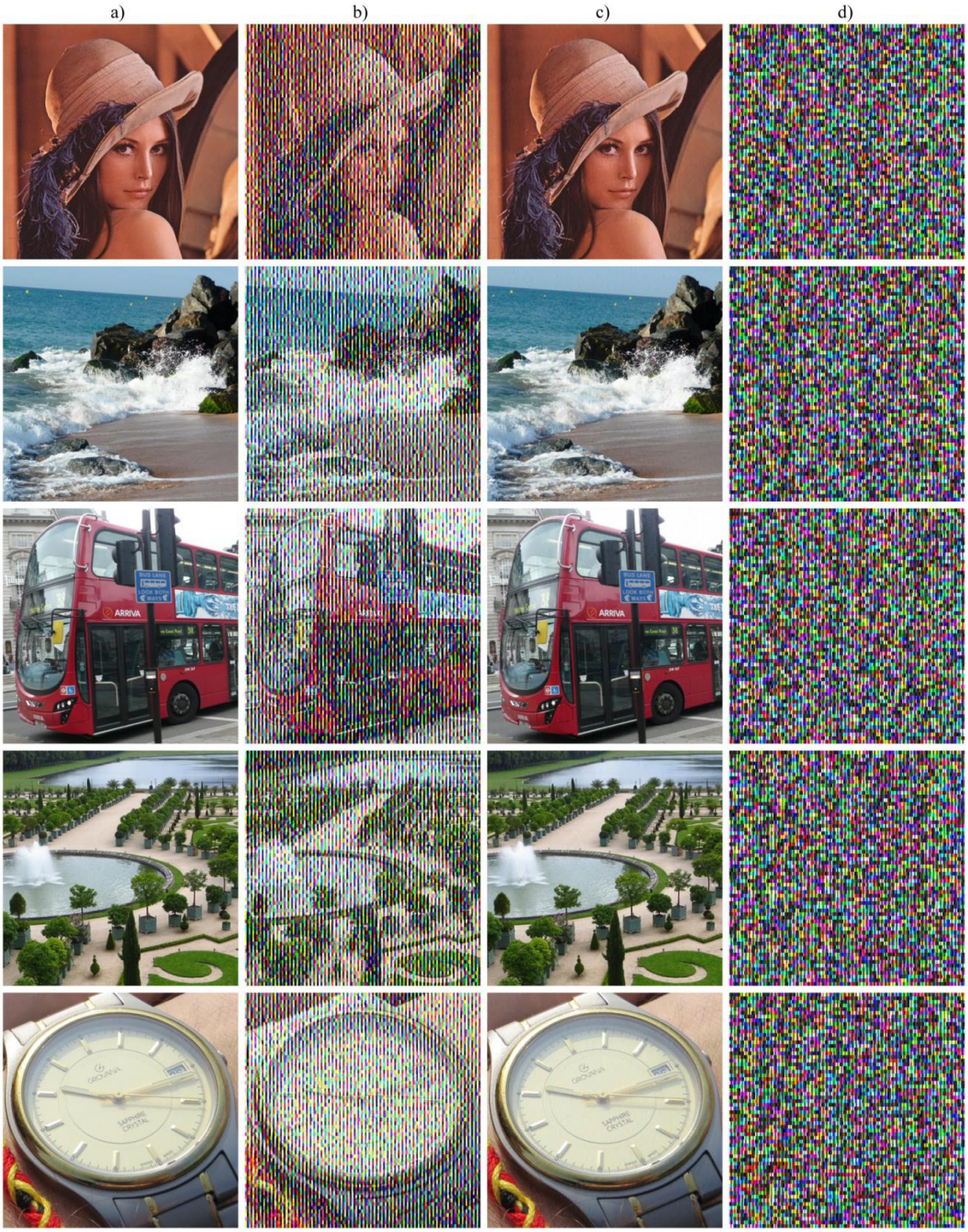
Rys. 9.2. Metoda A: przykładowe obrazy w postaci a) oryginalnej, b) zaszyfrowanej, c) odszyfrowanej z osadzonym fingerprintem oraz d) fingerprint wydobyty z obrazu odszyfrowanego, dla szyfrowania jednego współczynnika DCT w każdym bloku, opracowanie własne.



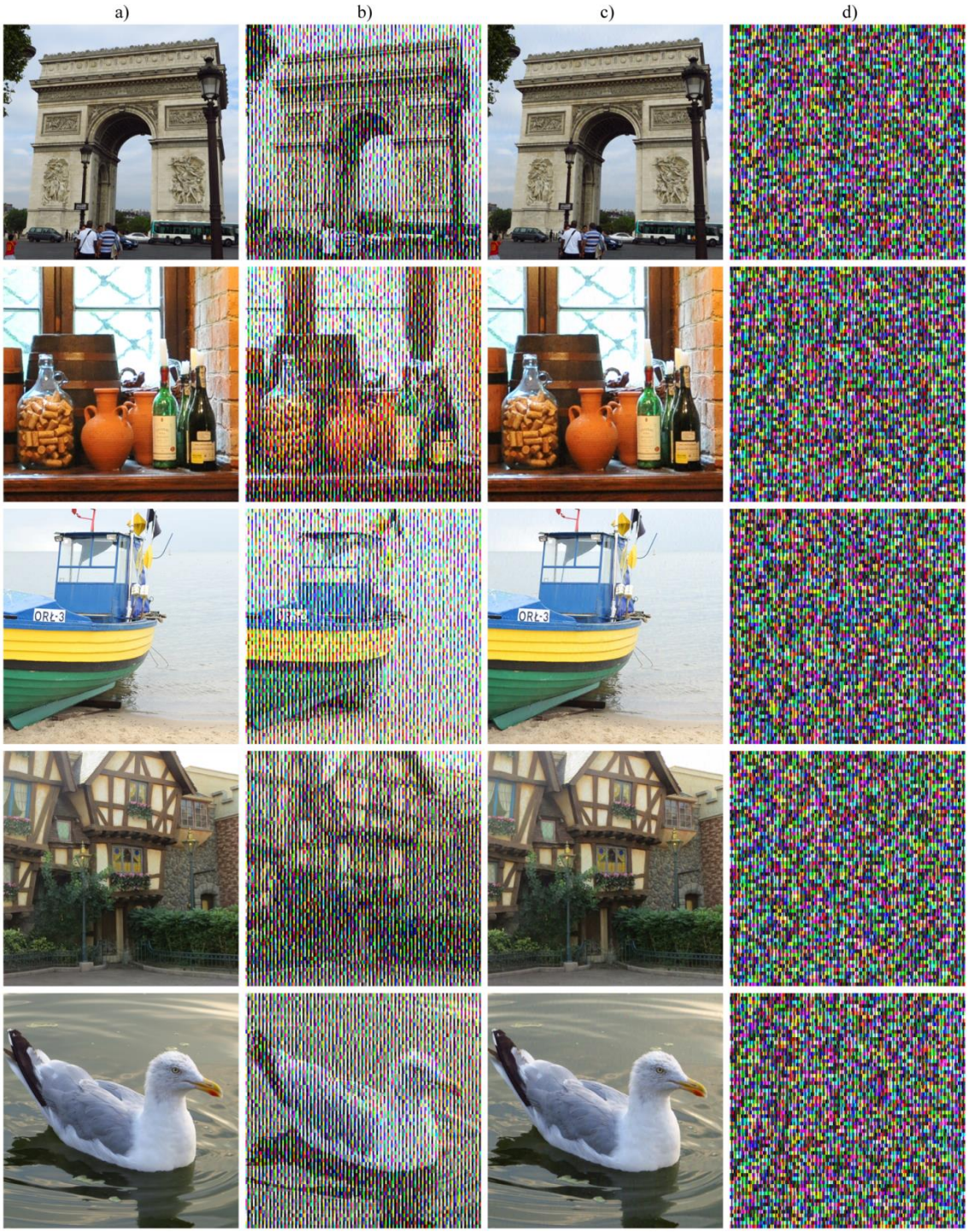
Rys. 9.3. Metoda A: przykładowe obrazy w postaci a) oryginalnej, b) zaszyfrowanej, c) odszyfrowanej z osadzonym fingerprintem oraz d) fingerprint wydobyty z obrazu odszyfrowanego, dla szyfrowania 16 współczynników DCT w każdym bloku, opracowanie własne.



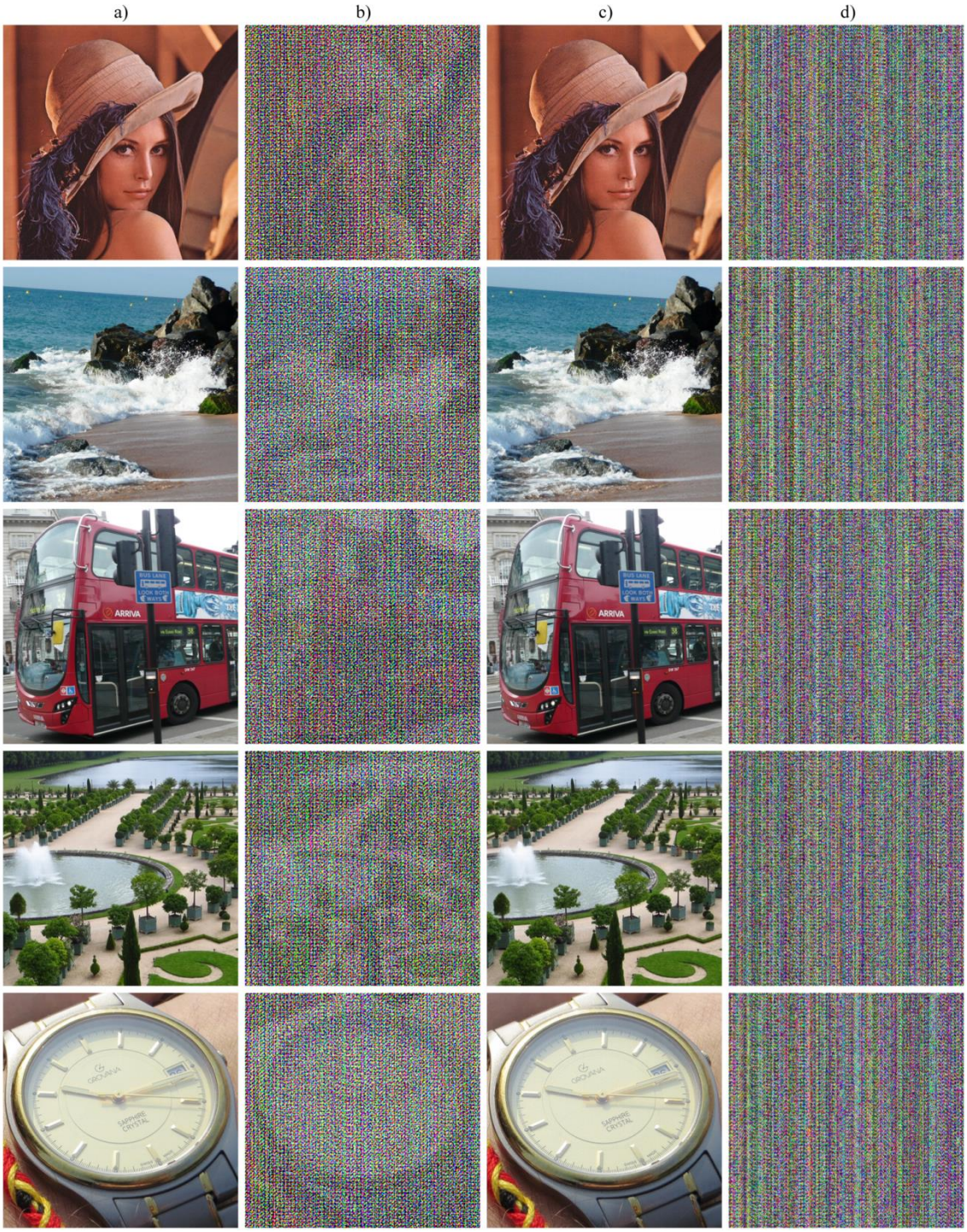
Rys. 9.4. Metoda A: przykładowe obrazy w postaci a) oryginalnej, b) zaszyfrowanej, c) odszyfrowanej z osadzonym fingerprintem oraz d) fingerprint wydobyty z obrazu odszyfrowanego, dla szyfrowania 16 współczynników DCT w każdym bloku, opracowanie własne.



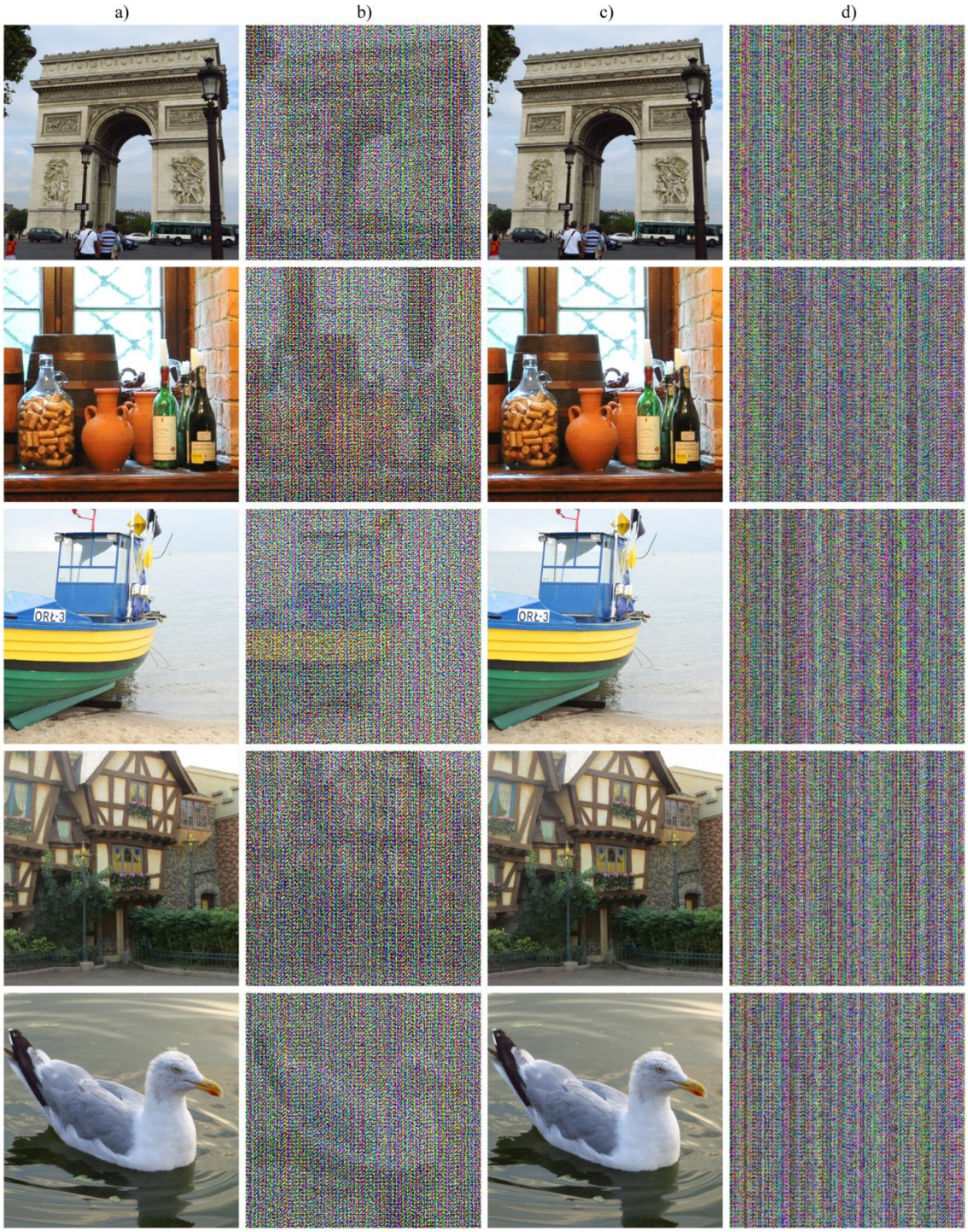
Rys. 9.5. Metoda B: przykładowe obrazy w postaci a) oryginalnej, b) zaszyfrowanej, c) odszyfrowanej z osadzonym fingerprintem oraz d) fingerprint wydobyty z obrazu odszyfrowanego, dla szyfrowania jednego współczynnika DCT w każdym bloku, opracowanie własne.



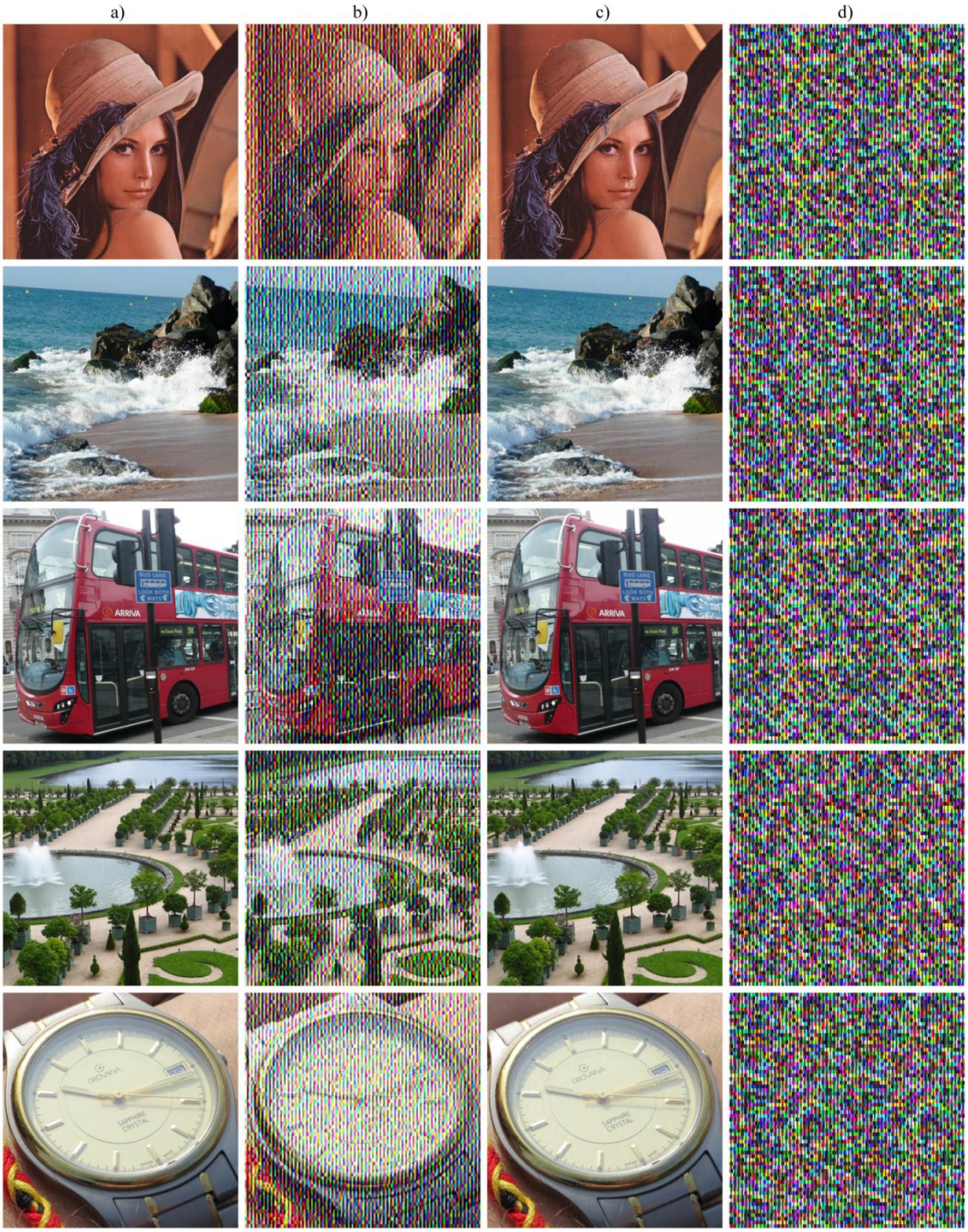
Rys. 9.6. Metoda B: przykładowe obrazy w postaci a) oryginalnej, b) zaszyfrowanej, c) odszyfrowanej z osadzonym fingerprintem oraz d) fingerprint wydobyty z obrazu odszyfrowanego, dla szyfrowania jednego współczynnika DCT w każdym bloku, opracowanie własne.



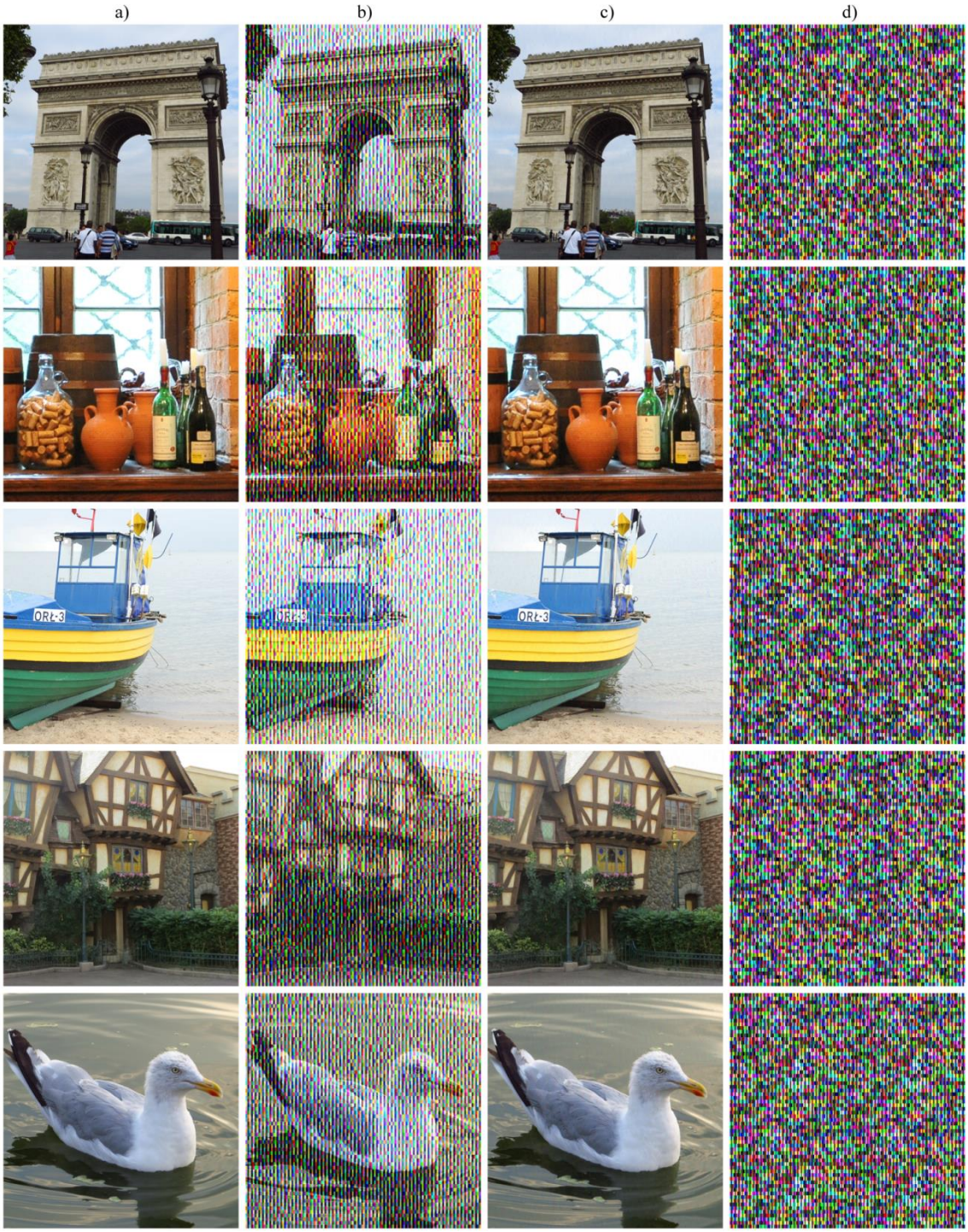
Rys. 9.7. Metoda B: przykładowe obrazy w postaci a) oryginalnej, b) zaszyfrowanej, c) odszyfrowanej z osadzonym fingerprintem oraz d) fingerprint wydobyty z obrazu odszyfrowanego, dla szyfrowania 16 współczynników DCT w każdym bloku, opracowanie własne.



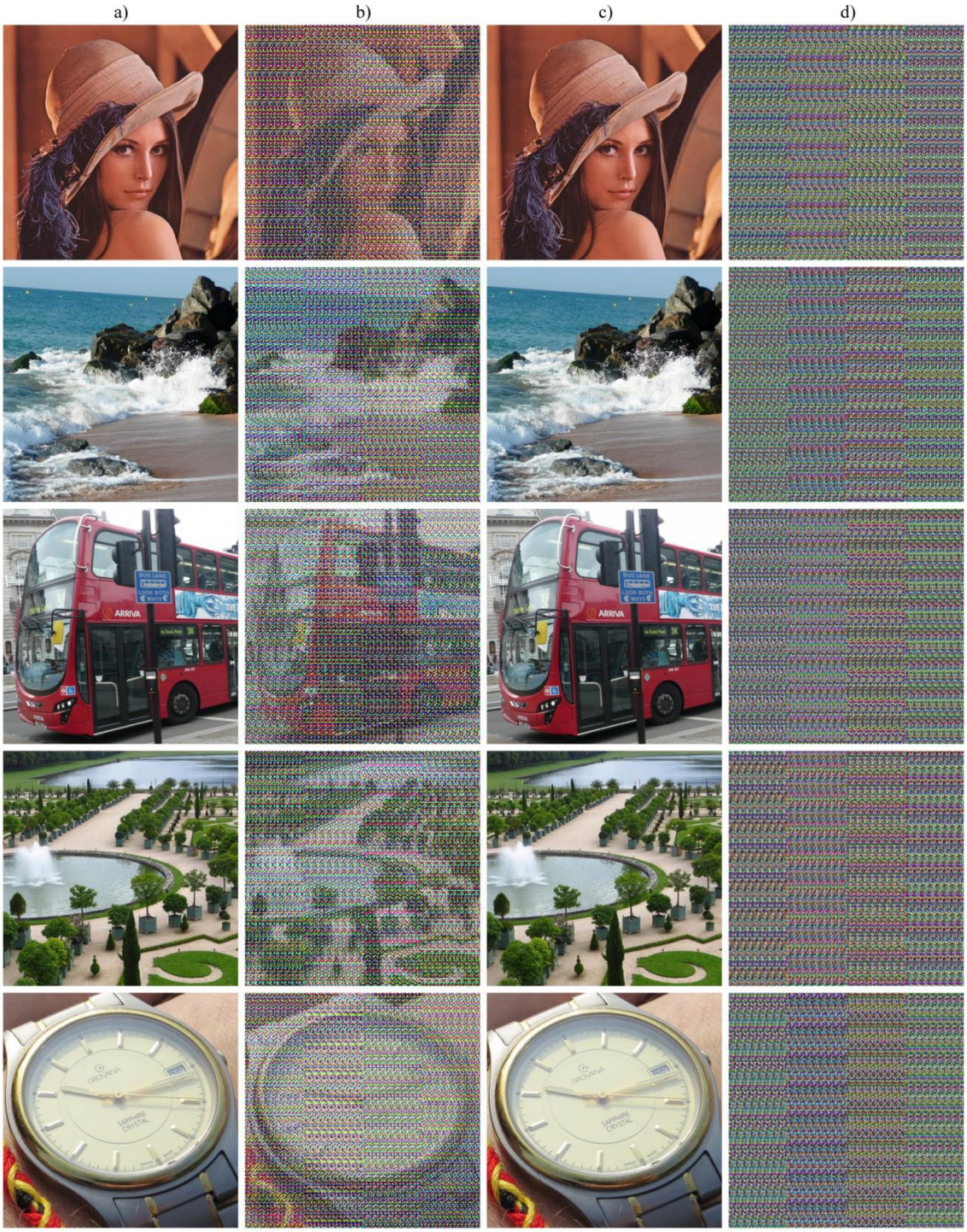
Rys. 9.8. Metoda B: przykładowe obrazy w postaci a) oryginalnej, b) zaszyfrowanej, c) odszyfrowanej z osadzonym fingerprintem oraz d) fingerprint wydobyty z obrazu odszyfrowanego, dla szyfrowania 16 współczynników DCT w każdym bloku, opracowanie własne.



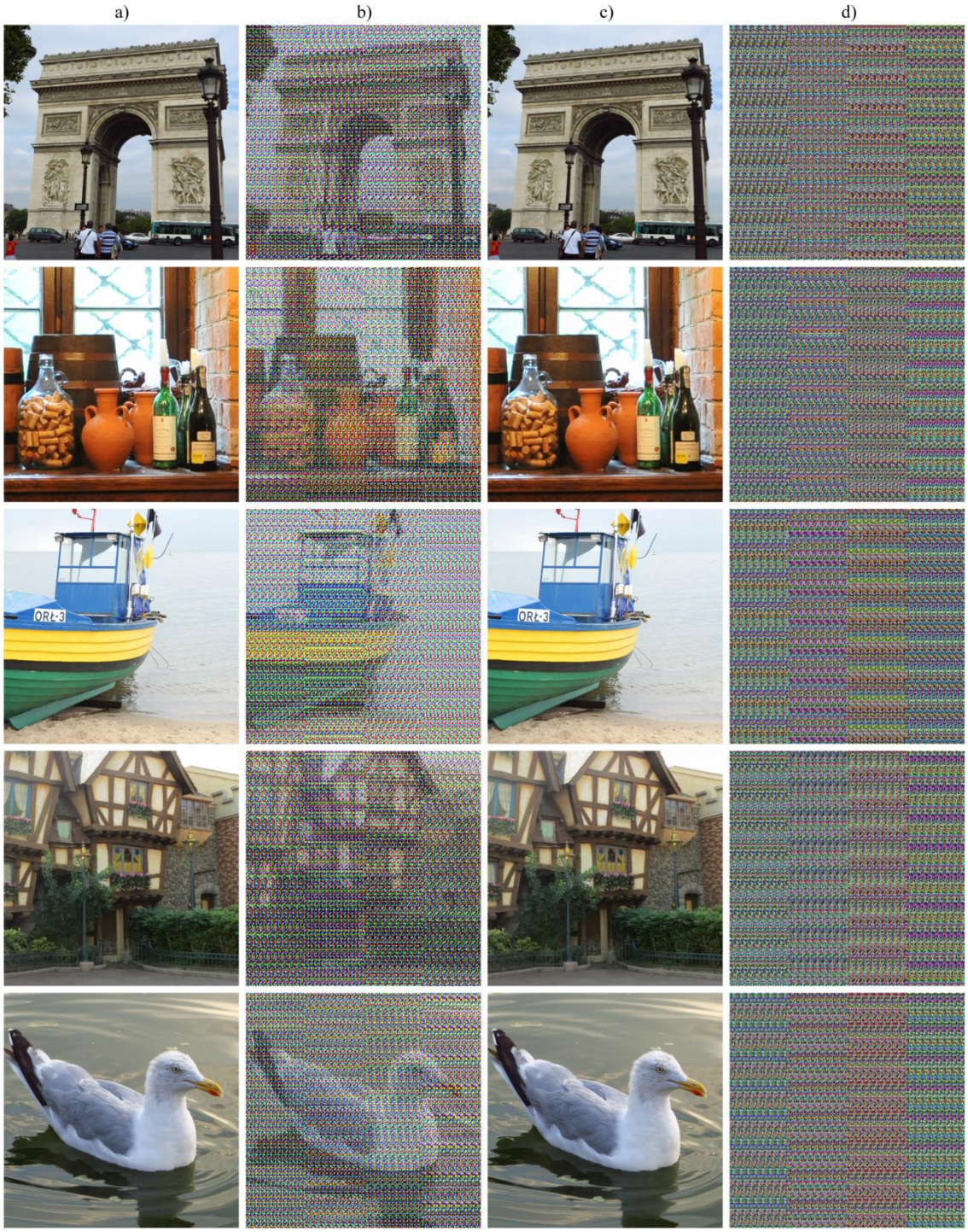
Rys. 9.9. Metoda C: przykładowe obrazy w postaci a) oryginalnej, b) zaszyfrowanej, c) odszyfrowanej z osadzonym fingerprintem oraz d) fingerprint wydobyty z obrazu odszyfrowanego, dla szyfrowania jednego współczynnika DCT w każdym bloku, opracowanie własne.



Rys. 9.10. Metoda C: przykładowe obrazy w postaci a) oryginalnej, b) zaszyfrowanej, c) odszyfrowanej z osadzonym fingerprintem oraz d) fingerprint wydobyty z obrazu odszyfrowanego, dla szyfrowania jednego współczynnika DCT w każdym bloku, opracowanie własne.



Rys. 9.11. Metoda C: przykładowe obrazy w postaci a) oryginalnej, b) zaszyfrowanej, c) odszyfrowanej z osadzonym fingerprintem oraz d) fingerprint wydobyty z obrazu odszyfrowanego, dla szyfrowania 16 współczynników DCT w każdym bloku, opracowanie własne.



Rys. 9.12. Metoda C: przykładowe obrazy w postaci a) oryginalnej, b) zaszyfrowanej, c) odszyfrowanej z osadzonym fingerprintem oraz d) fingerprint wydobyty z obrazu odszyfrowanego, dla szyfrowania 16 współczynników DCT w każdym bloku, opracowanie własne.

Następnie, zostały przeprowadzone obiektywne pomiary niezauważalności fingerprintów, przy wykorzystaniu szczytowego stosunku sygnału do szumu PSNR (ang. *Peak Signal-to-Noise Ratio*). Założono, że fingerprint jest osadzony w sposób niezauważalny, jeżeli PSNR dla rozpatrywanej składowej oznakowanego obrazu wynosi 40 dB lub więcej.

Dla metody A, przy szyfrowaniu jednego współczynnika na blok, wartość średnia PSNR dla składowej luminancji wynosi 46,641 dB z przedziałem ufności o szerokości 0,035

dB dla poziomu ufności 95% oraz liczebności próby 25000. Przy szyfrowaniu 16 współczynników na blok, wartość średnia PSNR wynosi 41,602 dB z przedziałem ufności o szerokości 0,106 dB dla poziomu ufności 95% oraz liczebności próby 25000.

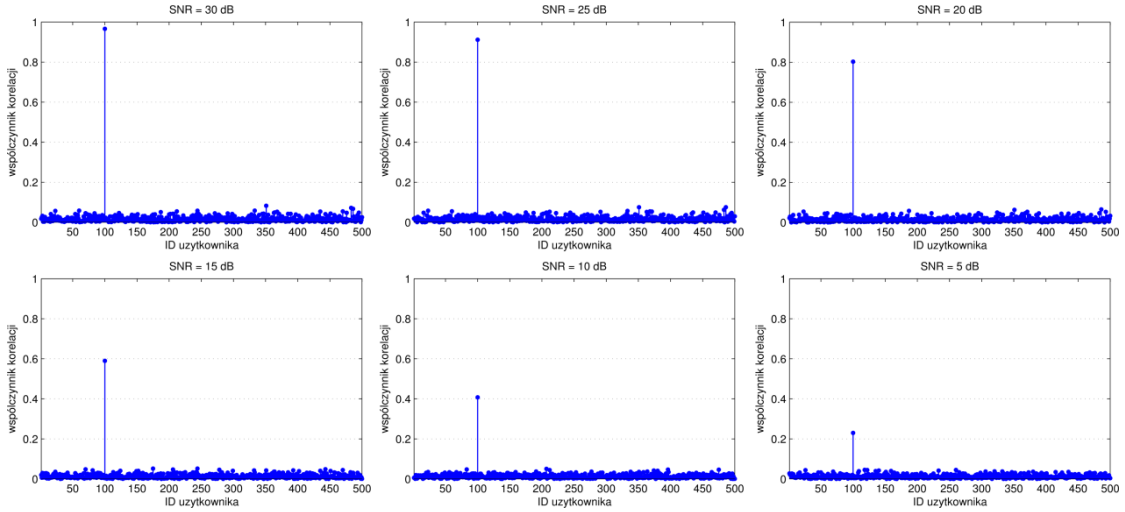
Dla metody B, przy szyfrowaniu jednego współczynnika na blok, wartość średnia PSNR dla składowej czerwonej wynosi 40,357 dB, dla zielonej 40,370 dB, a dla niebieskiej 40,368 dB z przedziałami ufności o szerokości 0,0043 dB dla każdego koloru, dla poziomu ufności 95% oraz liczebności próby 25000. Przy szyfrowaniu 16 współczynników na blok, wartość średnia PSNR dla składowej czerwonej wynosi 40,633 dB, dla zielonej 40,647 dB, a dla niebieskiej 40,624 z przedziałem ufności o szerokości 0,0188 dB dla każdego koloru, dla poziomu ufności 95% oraz liczebności próby 1000.

Dla metody C, przy szyfrowaniu jednego współczynnika na blok, wartość średnia PSNR dla składowej czerwonej wynosi 40,790 dB, dla zielonej 40,780 dB, a dla niebieskiej 40,788 dB z przedziałami ufności o szerokości 0,0046 dB dla każdego koloru, dla poziomu ufności 95% oraz liczebności próby 25000. Przy szyfrowaniu 16 współczynników na blok, wartość średnia PSNR dla składowej czerwonej wynosi 41,140 dB, dla zielonej 41,096, a dla niebieskiej 41.167 dB z przedziałem ufności o szerokości 0,171 dB dla składowej czerwonej, 0,187 dB dla składowej zielonej, oraz 0,148 dB dla składowej zielonej, dla poziomu ufności 95% oraz liczebności próby 1000. Na podstawie powyższych wyników pomiarów stwierdzono, że niezauważalność osadzonych fingerprintów została osiągnięta we wszystkich trzech badanych metodach.

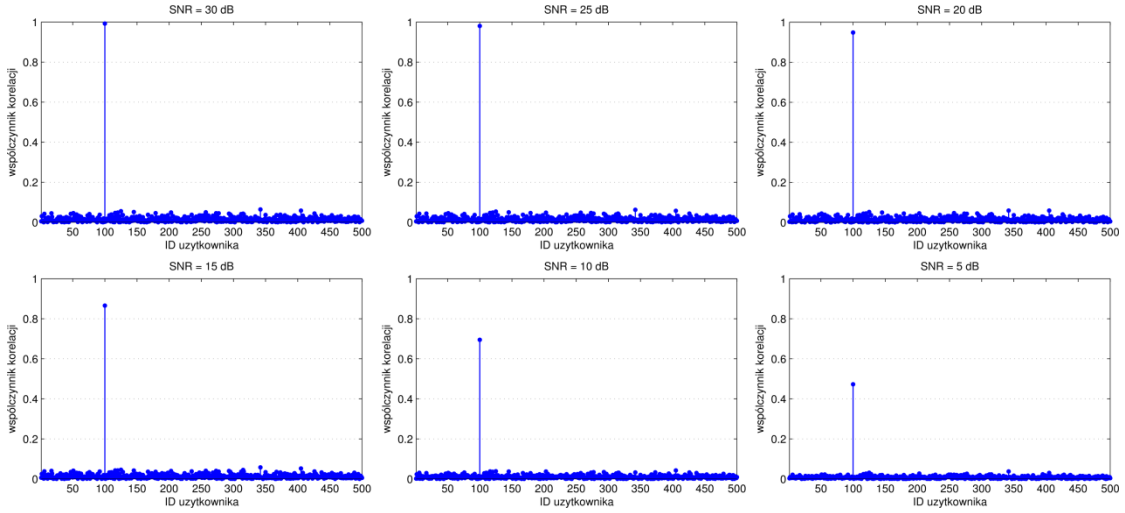
9.3. Odporność na szum

Dla wszystkich zaproponowanych metod JFD, zbadano odporność na atak poprzez dodanie szumu do obrazu oznakowanego fingerprintem. W tym ataku, jeden użytkownik (o identyfikatorze 100) generuje piracką kopię obrazu, będącą zaszumionym obrazem odszyfrowanym tego użytkownika, przy jednoczesnym zachowaniu zadowalającej jakości wizualnej. Szum był dodawany do składowej luminancji w przypadku metody A, lub do składowej każdego koloru RGB w przypadku metody B i C, ponieważ tam były osadzone fingerprinty. Dodawanym szumem był szum gaussowski o wartości średniej równej 0 oraz wariancji zależnej od pożądanego stosunku sygnału do szumu SNR (ang. *Signal-to-Noise Ratio*). Dla każdej metody JFD, w każdej z 50 symulacji, zrealizowano 6 takich ataków dla różnego poziomu zaszumienia określonego przez SNR równy 30 dB, 25 dB, 20 dB, 15 dB, 10 dB oraz 5 dB.

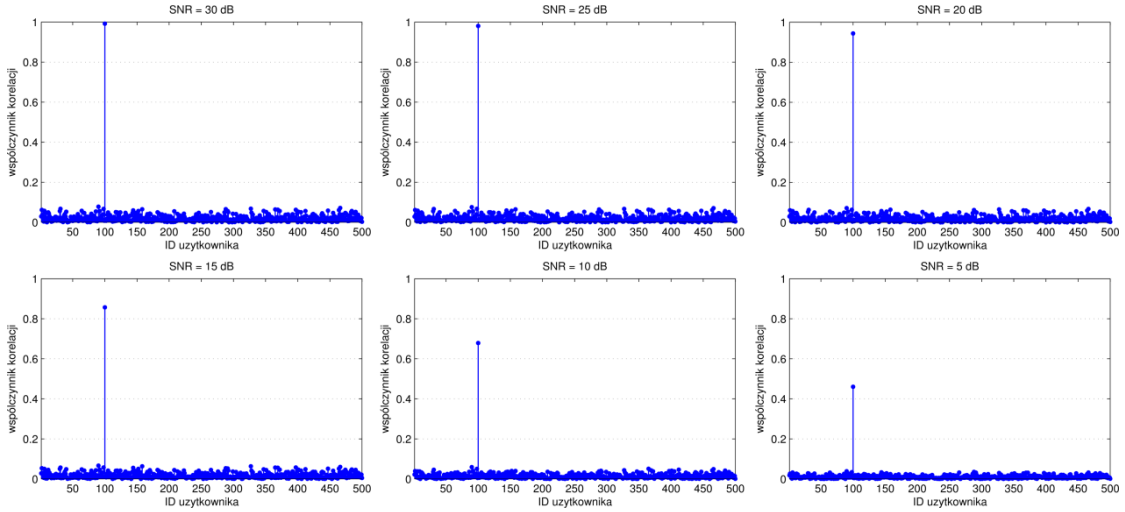
Przykłady identyfikacji pirata w przypadku ataku poprzez dodanie szumu, dla określonego SNR, dla metody A przedstawiono na rysunku 9.13., dla metody B na rysunku 9.14., dla metody C na rysunku 9.15. Można zaobserwować, że w każdym ataku współczynnik korelacji fingerprintu wydobytego z pirackiej kopii oraz fingerprintu użytkownika z ID = 100 (pirata) ma znacznie większą wartość niż współczynniki korelacji dla fingerprintów pozostałych użytkowników. Przy takich właściwościach, bardzo łatwo ustalić próg detekcji do identyfikacji winnego użytkownika. W każdej symulacji, dla wszystkich badanych metod, osiągnięto bezbłędną identyfikację pirata, tj. $P_{TP} = 1$ oraz $P_{FP} = 0$, dla wszystkich rozpatrywanych wartości SNR. W każdej symulacji poprawnie zidentyfikowano pirata bez oskarżania niewinnych użytkowników, co oznacza, że wszystkie proponowane metody są odporne na szum.



Rys. 9.13. Metoda A: identyfikacja pirata w przypadku ataku poprzez dodanie szumu dla SNR równego 30 dB, 25 dB, 20 dB, 15 dB, 10 dB oraz 5 dB, opracowanie własne.



Rys. 9.14. Metoda B: identyfikacja pirata w przypadku ataku poprzez dodanie szumu dla SNR równego 30 dB, 25 dB, 20 dB, 15 dB, 10 dB oraz 5 dB, opracowanie własne.

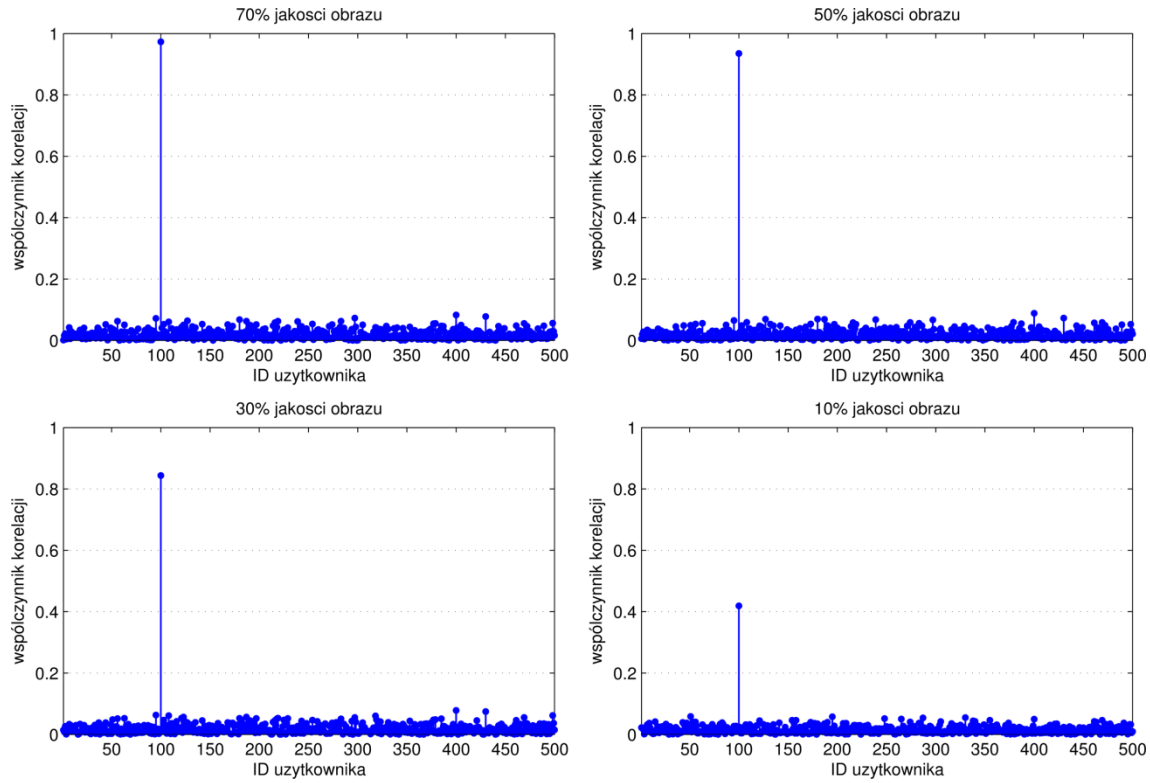


Rys. 9.15. Metoda C: identyfikacja pirata w przypadku ataku poprzez dodanie szumu dla SNR równego 30 dB, 25 dB, 20 dB, 15 dB, 10 dB oraz 5 dB, opracowanie własne.

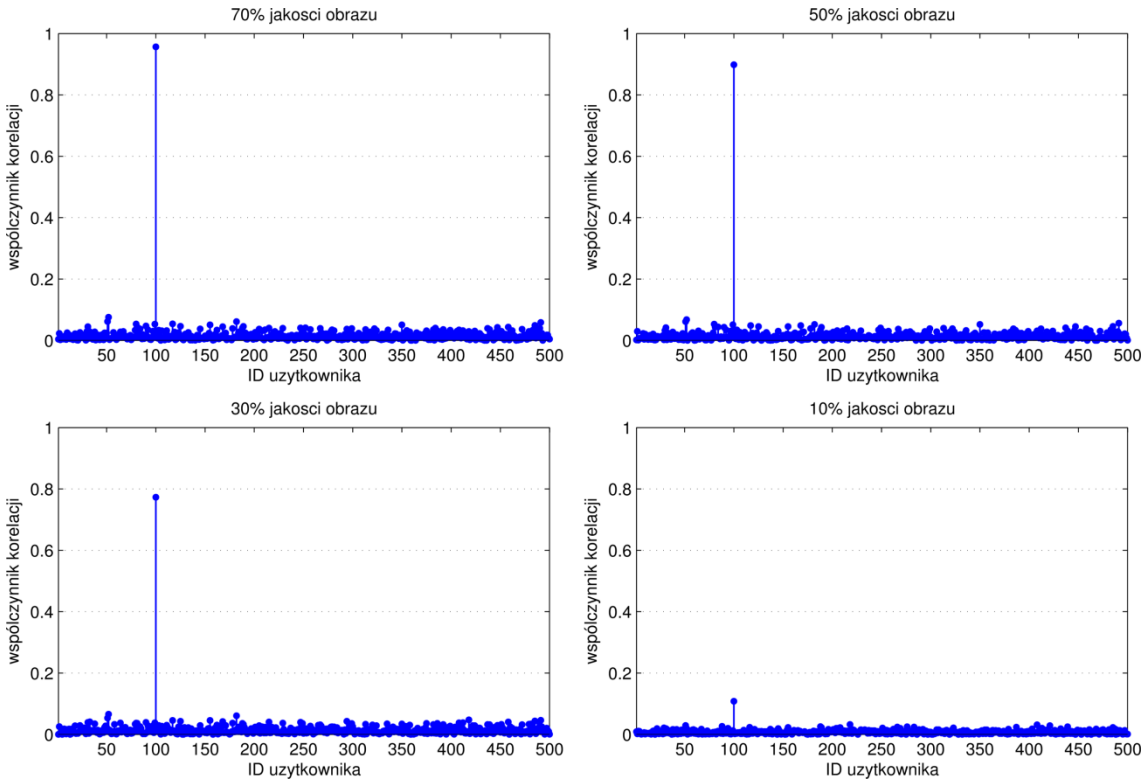
9.4. Odporność na kompresję

Dla wszystkich zaproponowanych metod JFD, zbadano odporność na atak poprzez kompresję obrazu oznakowanego fingerprintem. W tym ataku, jeden użytkownik (o identyfikatorze 100) generuje piracką kopię obrazu, będącą skompresowanym obrazem odszyfrowanym tego użytkownika, przy jednoczesnym zachowaniu zadowalającej jakości wizualnej. Wykorzystanym algorytmem kompresji był standard kompresji JPEG. Dla każdej badanej metody JFD, w każdej z 50 symulacji, zrealizowano 4 takie ataki dla różnych poziomów jakości, tzn. 70%, 50%, 30% oraz 10% jakości obrazu skompresowanego.

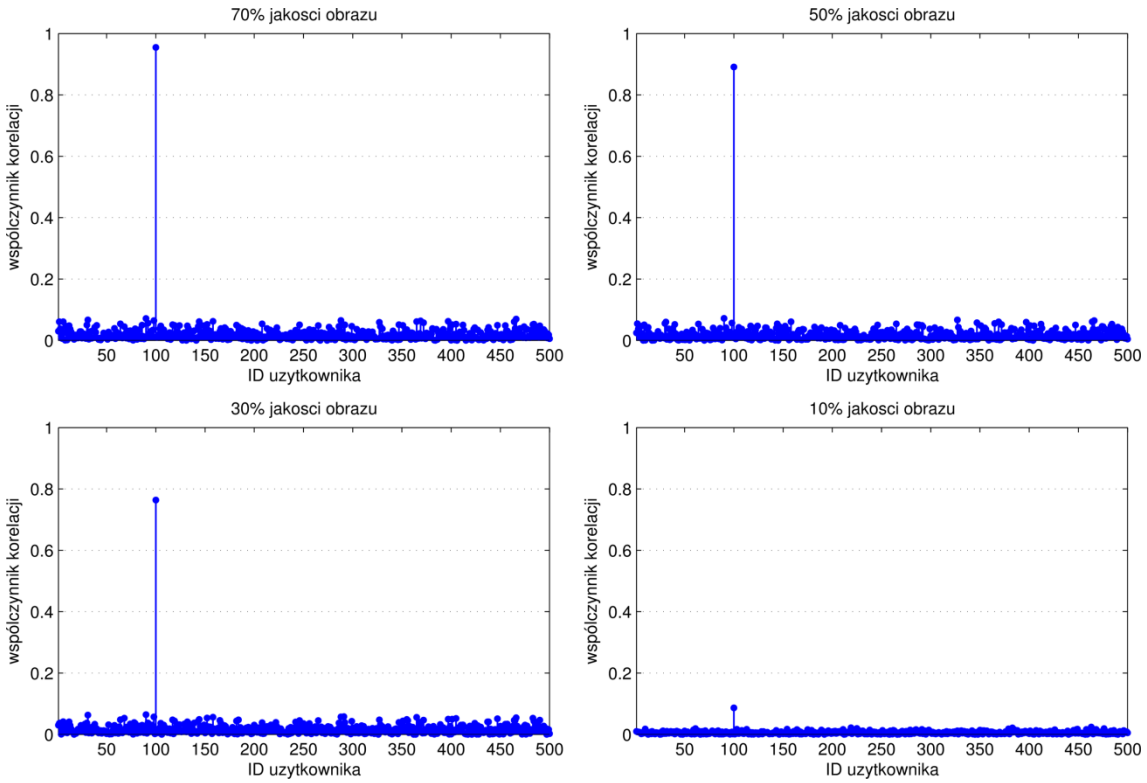
Przykłady identyfikacji pirata w przypadku ataku poprzez kompresję, dla określonej jakości obrazu skompresowanego, dla metody A przedstawiono na rysunku 9.16., dla metody B na rysunku 9.17., dla metody C na rysunku 9.18. Można zaobserwować, że w każdym ataku współczynnik korelacji fingerprintu wydobytego z pirackiej kopii oraz fingerprintu użytkownika z ID = 100 (pirata) ma znacznie większą wartość niż współczynniki korelacji dla fingerprintów pozostałych użytkowników. Dla metody B i C, w przypadku 10% jakości obrazu, różnica między wartością współczynnika koleracji dla pirata, a wartościami dla pozostałych użytkowników wynosi tylko ~ 0,08 – 0,1, ale nadal jest to wartość kilkukrotnie większa. Przy takich właściwościach, bardzo łatwo ustalić próg detekcji do identyfikacji winnego użytkownika. W każdej symulacji, dla wszystkich badanych metod, osiągnięto bezbłędną identyfikację pirata, tj. $P_{TP} = 1$ oraz $P_{FP} = 0$, dla wszystkich rozpatrywanych wartości SNR. W każdej symulacji poprawnie zidentyfikowano pirata bez oskarżania niewinnych użytkowników, co oznacza, że wszystkie proponowane metody są odporne na kompresję.



Rys. 9.16. Metoda A: przykłady identyfikacji pirata w przypadku ataku poprzez kompresję dla 70 %, 50%, 30% oraz 10% jakości obrazu skompresowanego, opracowanie własne.



Rys. 9.17. Metoda B: przykłady identyfikacji pirata w przypadku ataku poprzez kompresję dla 70 %, 50%, 30% oraz 10% jakości obrazu skompresowanego, opracowanie własne.



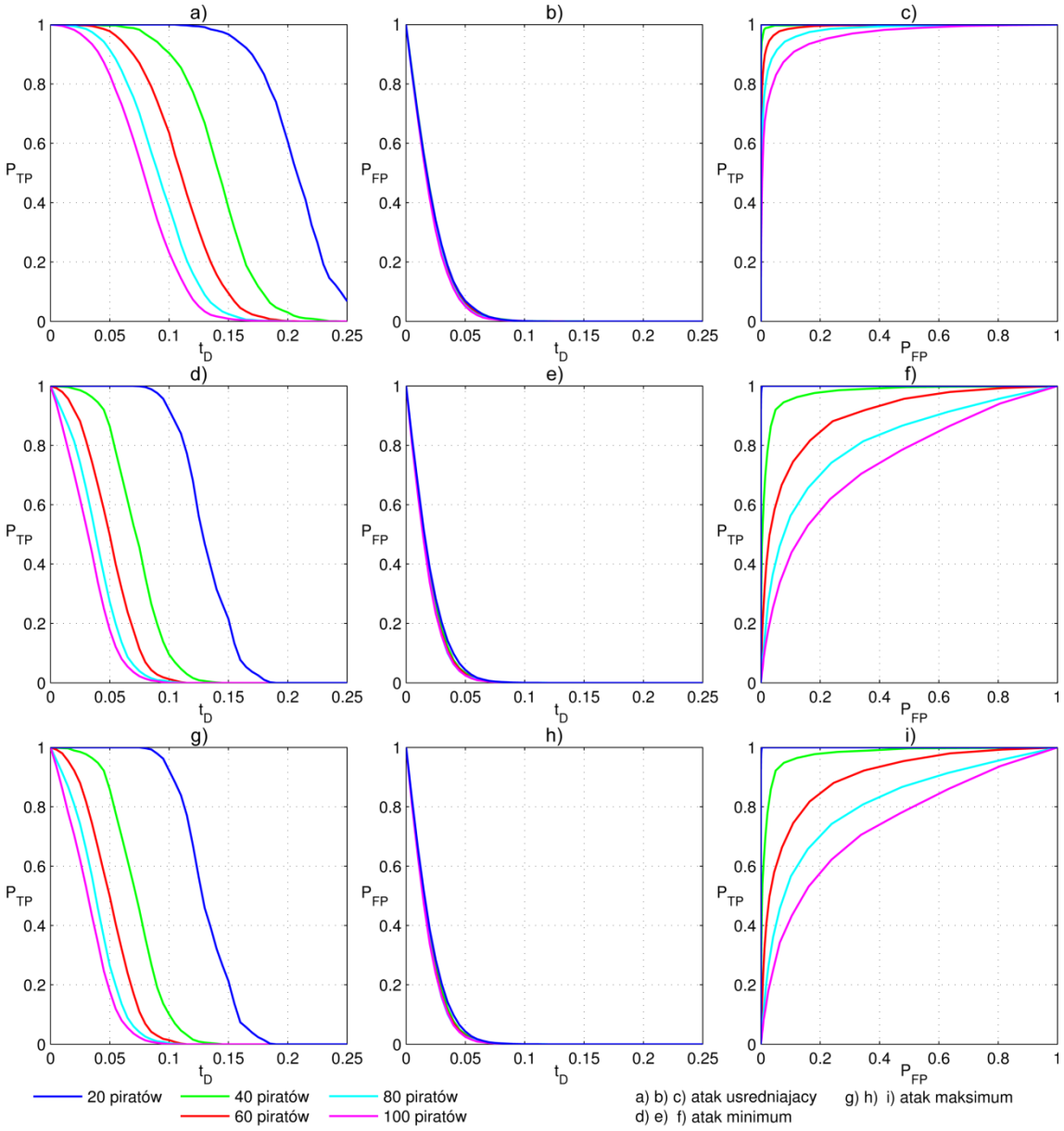
Rys. 9.18. Metoda C: przykłady identyfikacji pirata w przypadku ataku poprzez kompresję dla 70 %, 50%, 30% oraz 10% jakości obrazu skompresowanego, opracowanie własne.

9.5. Odporność na atak zmowy

Dla wszystkich zaproponowanych metod JFD, zbadano odporność na ataki zmowy. W tych atakach, grupa piratów generuje piracką kopię obrazu na podstawie swoich kopii obrazów, przy jednoczesnym zachowaniu zadowalającej jakości wizualnej. W badaniach uwzględniono 6 typów ataku zmowy: atak uśredniający, atak minimum, atak maksimum, atak medianowy, atak min-max oraz losowy atak różnicowy. Dla każdej badanej metody JFD, w każdej z 50 symulacji, zrealizowano 30 różnych ataków zmowy, tzn. w każdym z 6 typów ataku użyto 5 różnych liczebności piratów w zmowie: 20, 40, 60, 80 oraz 100 piratów. Podczas badań, rozpatrywano zarówno liniowe, jak i nieliniowe ataki zmowy. Jednakże, najwięcej uwagi należy poświęcić wynikom dotyczącym ataku uśredniającego, z powodów opisanych w rozdziale 3.2. Po pierwsze, atak ten jest sprawiedliwą zmową, gdyż energia fingerprintu każdego pirata jest zredukowana o taki sam współczynnik, zatem ryzyko wykrycia jest równomiernie rozłożone pomiędzy wszystkich uczestników zmowy. Po drugie, atak ten nie powoduje pogorszenia jakości wizualnej pirackiej kopii. Po trzecie, metody fingerprintingu opisywane w literaturze zazwyczaj wykorzystują tylko ten atak do badania odporności osadoznych fingerprintów, zatem odporność na atak uśredniający jest powszechnie używanym punktem odniesienia, dającym możliwość porównania z innymi metodami.

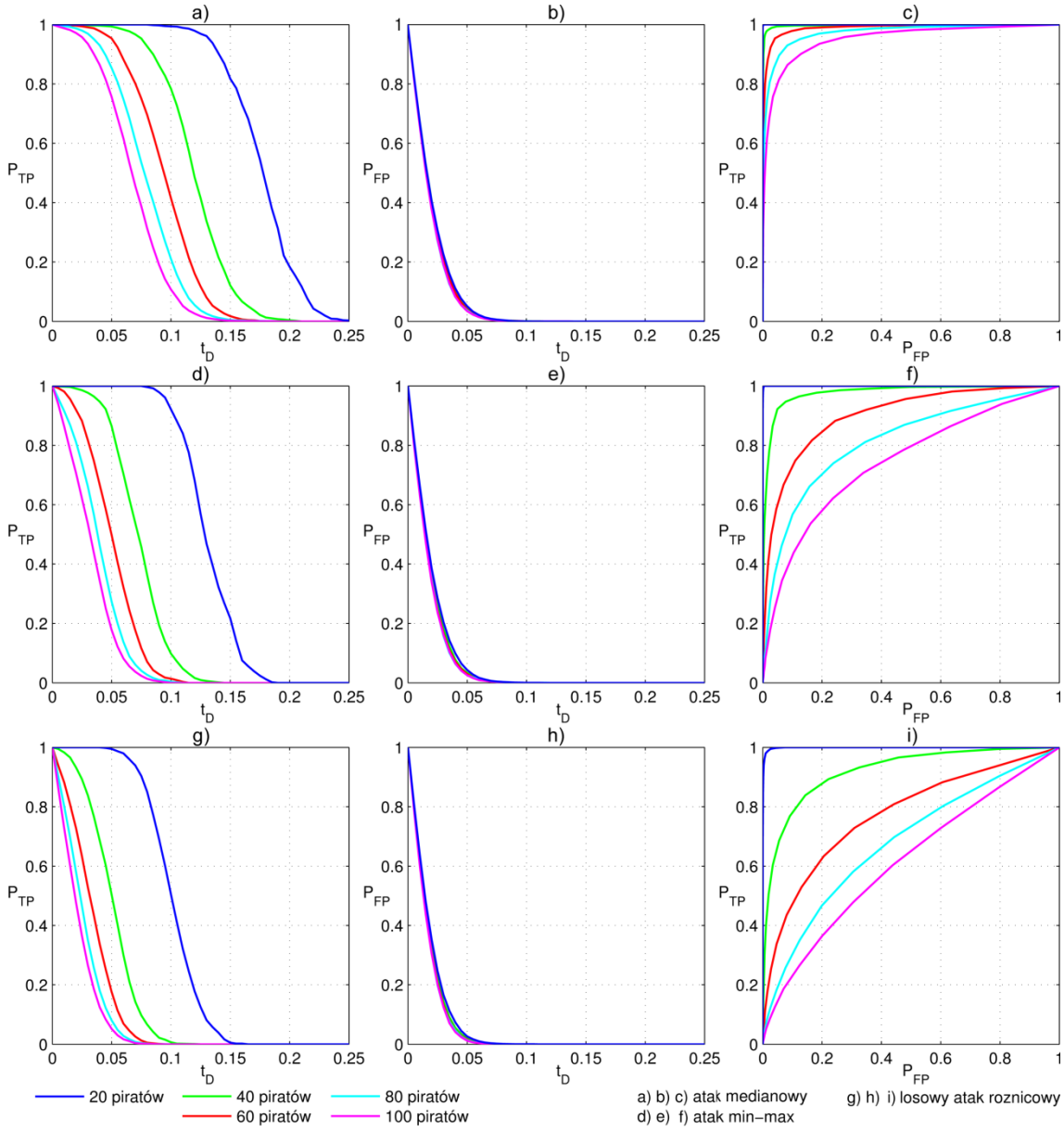
Wyniki badania odporności na ataki zmowy dla metody A przedstawiono na rysunkach 9.19. i 9.20., dla metody B na rysunkach 9.21. i 9.22., dla metody C na rysunkach 9.23. i 9.24. Wykresy w kolejnych wierszach odnoszą się do różnych typów ataków, tj. ataku uśredniającego, ataku minimum i ataku maximum na rysunkach 9.19., 9.21., 9.23. oraz ataku medianowego, ataku min-max i losowego ataku różnicowego na rysunkach 9.20., 9.22., 9.24. Różne kolory wykresów odnoszą się do różnej liczby piratów w zmowie, tj. 20, 40, 60, 80 i 100 piratów. Wykresy w pierwszej kolumnie przedstawiają zależność P_{TP} od progu detekcji t_D. Na tych wykresach, krzywa najbardziej wysunięta w prawą stronę oznacza większą skuteczność identyfikacji piratów, a dokładniej większą szansę poprawnej identyfikacji pirata. Wykresy w drugiej kolumnie przedstawiają zależność P_{FP} od progu t_D. Na tych wykresach, krzywa bardziej wysunięta w lewą stronę oznacza większą skuteczność identyfikacji piratów, a dokładniej mniejsze ryzyko oskarżenia niewinnych użytkowników. Należy zwrócić uwagę, że prawdopodobieństwo P_{FP} nie zmienia się znacząco wraz z rosnącą liczbą piratów, co jest pożądaną właściwością. Wykresy w trzeciej kolumnie przedstawiają krzywe ROC, czyli zależność P_{TP} od P_{FP}. Na tych wykresach, krzywa przebiegająca blisko lewego górnego rogu wykresu oznacza bardzo dużą skuteczność, natomiast krzywa bliska przekątnej oznacza niską skuteczność.

Wyniki badań świadczą o bardzo dużej odporności osadzonych fingerprintów na ataki zmowy. W celu określenia maksymalnej liczby piratów w zmowie, na którą proponowana metoda jest odporna, wprowadzono następującą definicję odporności na ataki zmowy. Jeżeli prawdopodobieństwo poprawnie pozytywnej decyzji wynosi $P_{TP} \geq 0{,}9$, przy ustalonym prawdopodobieństwie fałszywie pozytywnej decyzji wynoszącym $P_{FP} = 0{,}01$ oraz przy liczbie piratów C, to badana metoda jest odporna na atak zmowy realizowany przez C piratów. Natomiast, bezbłędna identyfikacja oznacza, że poprawnie zidentyfikowano wszystkich piratów, bez oskarżania żadnego niewinnego użytkownika.

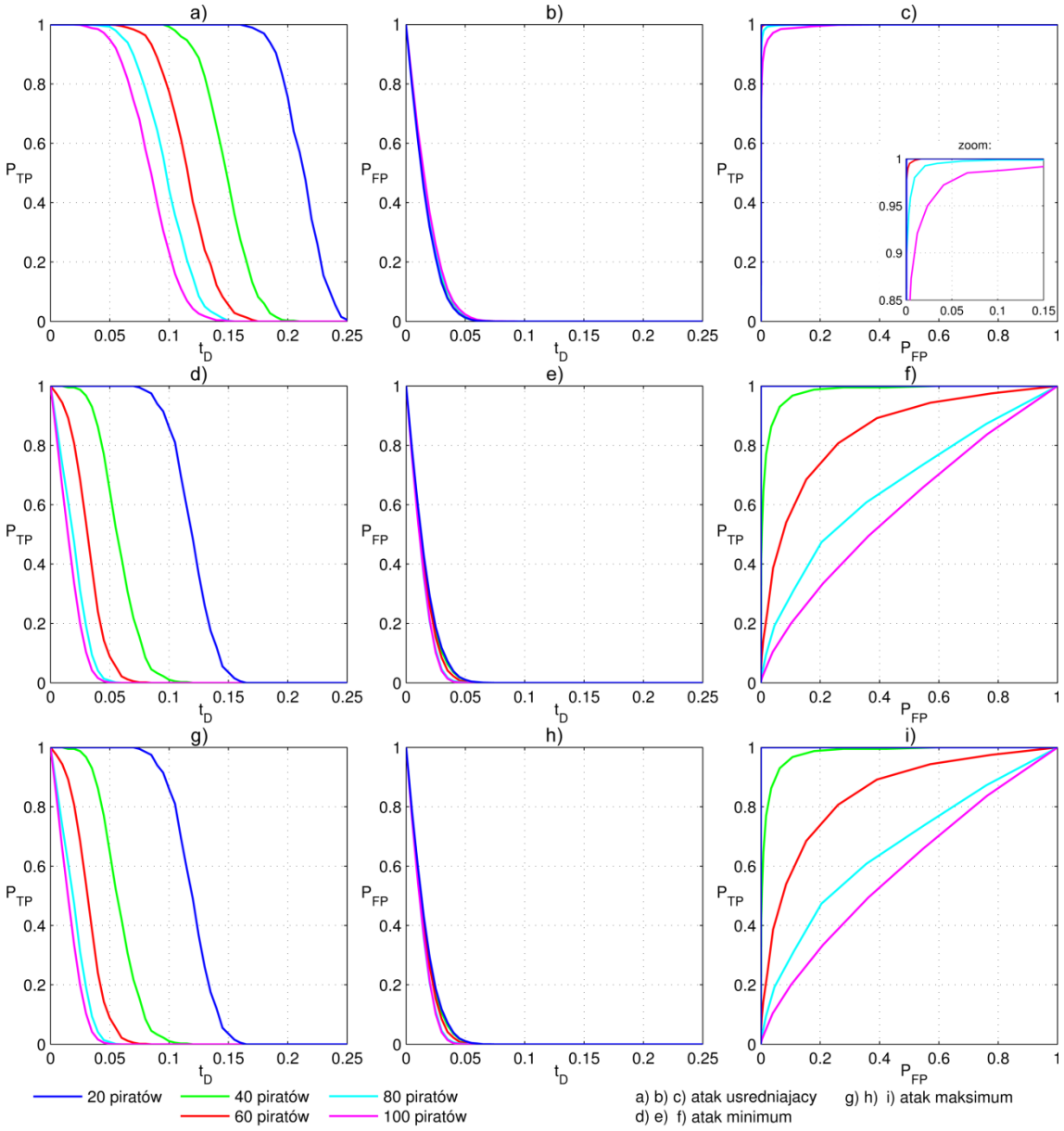


Rys. 9.19. Metoda A: odporność na ataki zmowy (atak uśredniający, atak minimum, atak maksimum) w przypadku zmowy 20, 40, 60, 80 i 100 piratów, opracowanie własne.

Dla metody A, wyniki badań odporności na atak zmowy metodą uśredniania są następujące. Przy założeniu, że akceptowalne prawdopodobieństwo oskarżenia niewinnego użytkownika wynosi $P_{FP} = 0{,}01$, prawdopodobieństwo identyfikacji pirata wynosi $P_{TP} = 1$ w przypadku zmowy 20 piratów, $P_{TP} = 0{,}979$ w przypadku 40 piratów, $P_{TP} = 0{,}899$ w przypadku 60 piratów, $P_{TP} = 0{,}741$ w przypadku 80 piratów, $P_{TP} = 0{,}628$ w przypadku 100 piratów. Na podstawie powyższych pomiarów i przyjętej definicji odporności można stwierdzić, że metoda A jest odporna na atak zmowy uśredniający realizowany przez 60 lub mniej piratów. Ponadto, w każdej symulacji osiągnięto bezbłędną identyfikację piratów, tj. $P_{TP} = 1$ oraz $P_{FP} = 0$, dla 20 piratów w przypadku ataku uśredniającego, ataku minimum, ataku maximum, ataku medianowego, oraz ataku min-max.

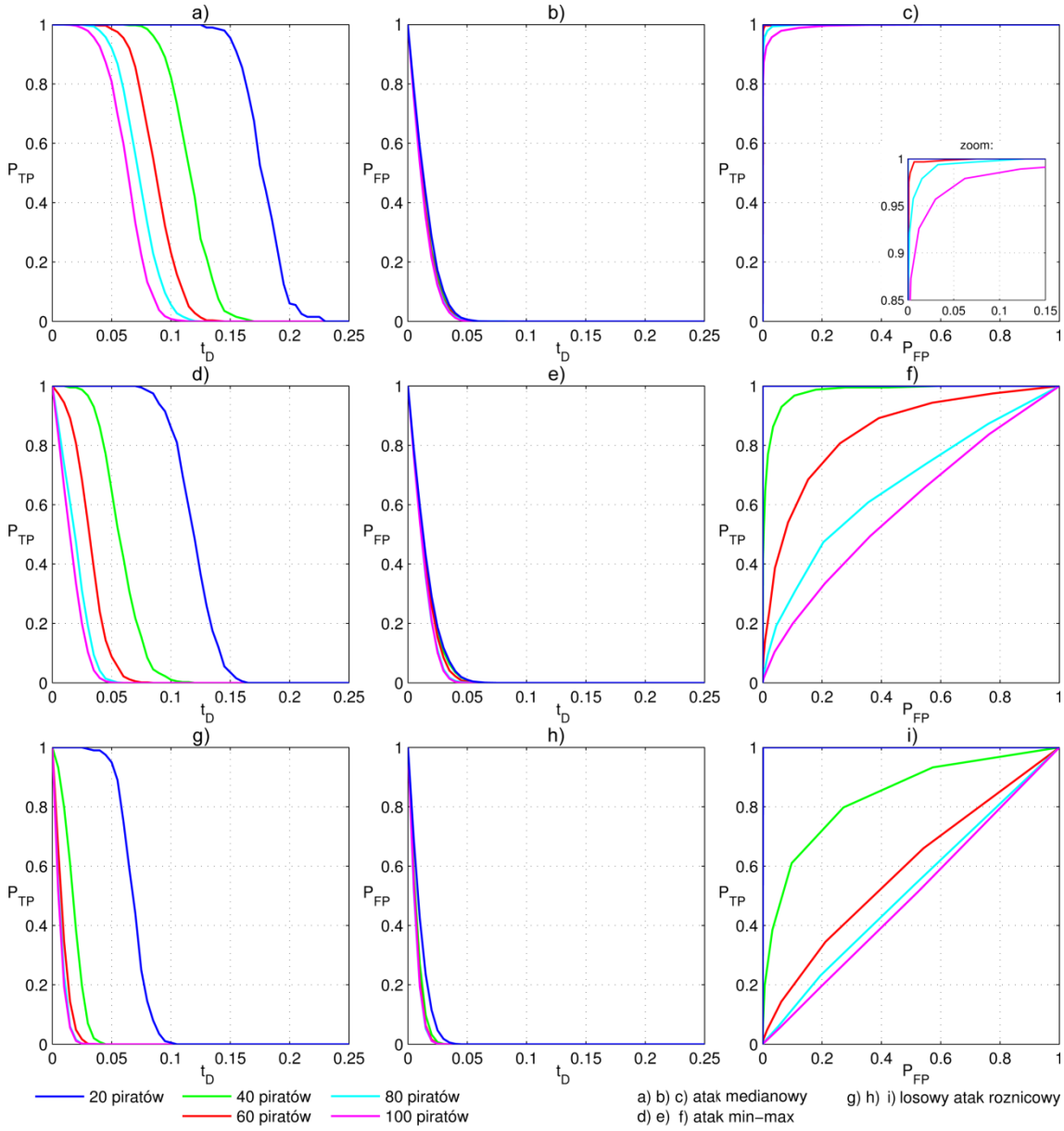


Rys. 9.20. Metoda A: odporność na ataki zmowy (atak medianowy, atak min-max, losowy atak różnicowy) w przypadku zmowy 20, 40, 60, 80 i 100 piratów, opracowanie własne.

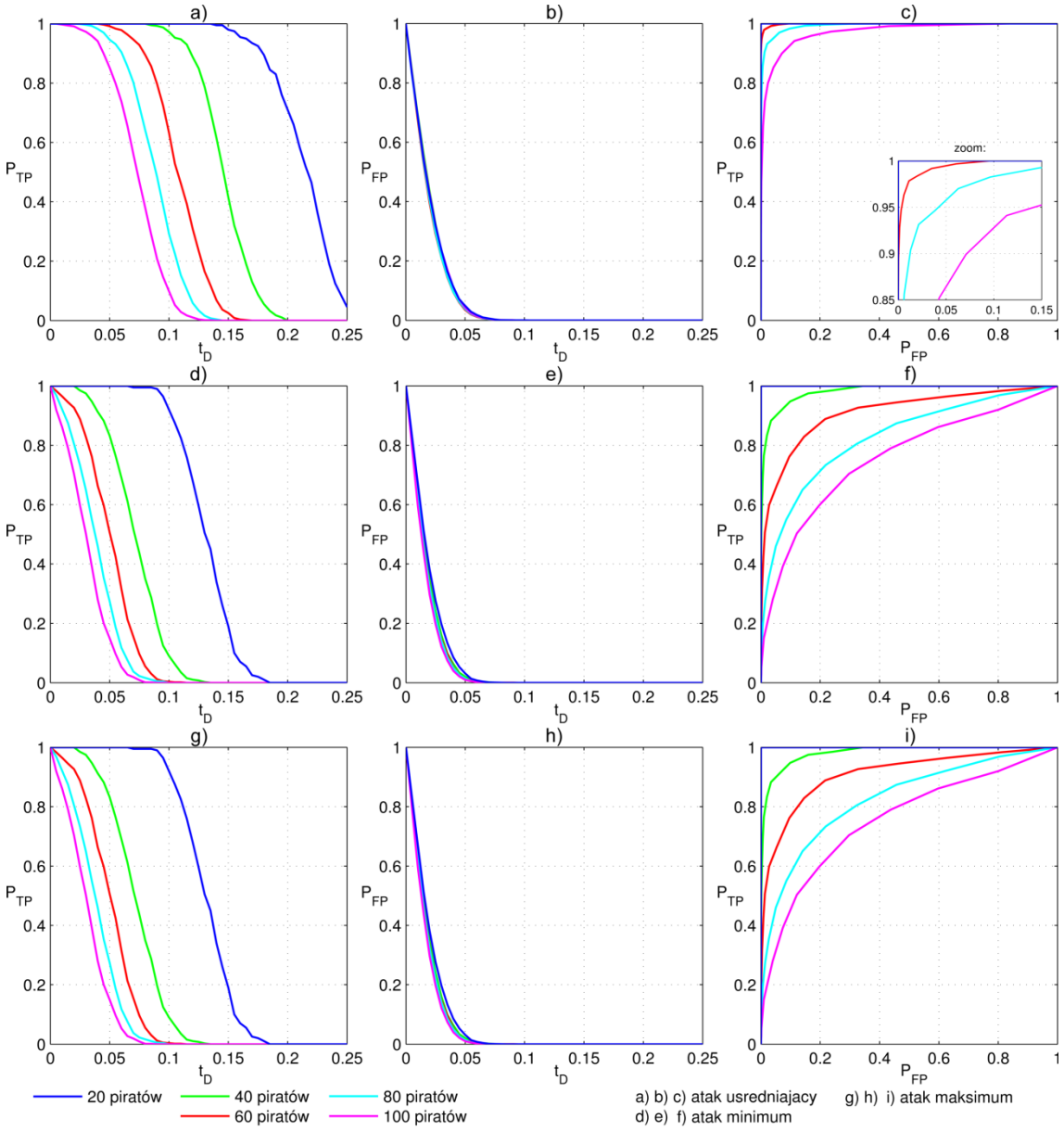


Rys. 9.21. Metoda B: odporność na ataki zmowy (atak uśredniający, atak minimum, atak maksimum) w przypadku zmowy 20, 40, 60, 80 i 100 piratów, opracowanie własne.

Dla metody B, wyniki badań odporności na atak zmowy metodą uśredniania są następujące. Przy założeniu, że akceptowalne prawdopodobieństwo oskarżenia niewinnego użytkownika wynosi $P_{FP} = 0{,}01$, prawdopodobieństwo identyfikacji pirata wynosi $P_{TP} = 1$ w przypadku zmowy 20 i 40 piratów, $P_{TP} = 0{,}999$ w przypadku 60 piratów, $P_{TP} = 0{,}981$ w przypadku 80 piratów, $P_{TP} = 0{,}903$ w przypadku 100 piratów. Na podstawie powyższych pomiarów i przyjętej definicji odporności można stwierdzić, że metoda B jest odporna na atak zmowy uśredniający, realizowany przez 100 lub mniej piratów. Ponadto, w każdej symulacji osiągnięto bezbłędną identyfikację piratów, tj. $P_{TP} = 1$ oraz $P_{FP} = 0$, dla 20 piratów w przypadku ataku minimum, ataku maximum, ataku min-max oraz losowego ataku różnicowego, a nawet dla 40 piratów w przypadku ataku uśredniającego oraz ataku medianowego.

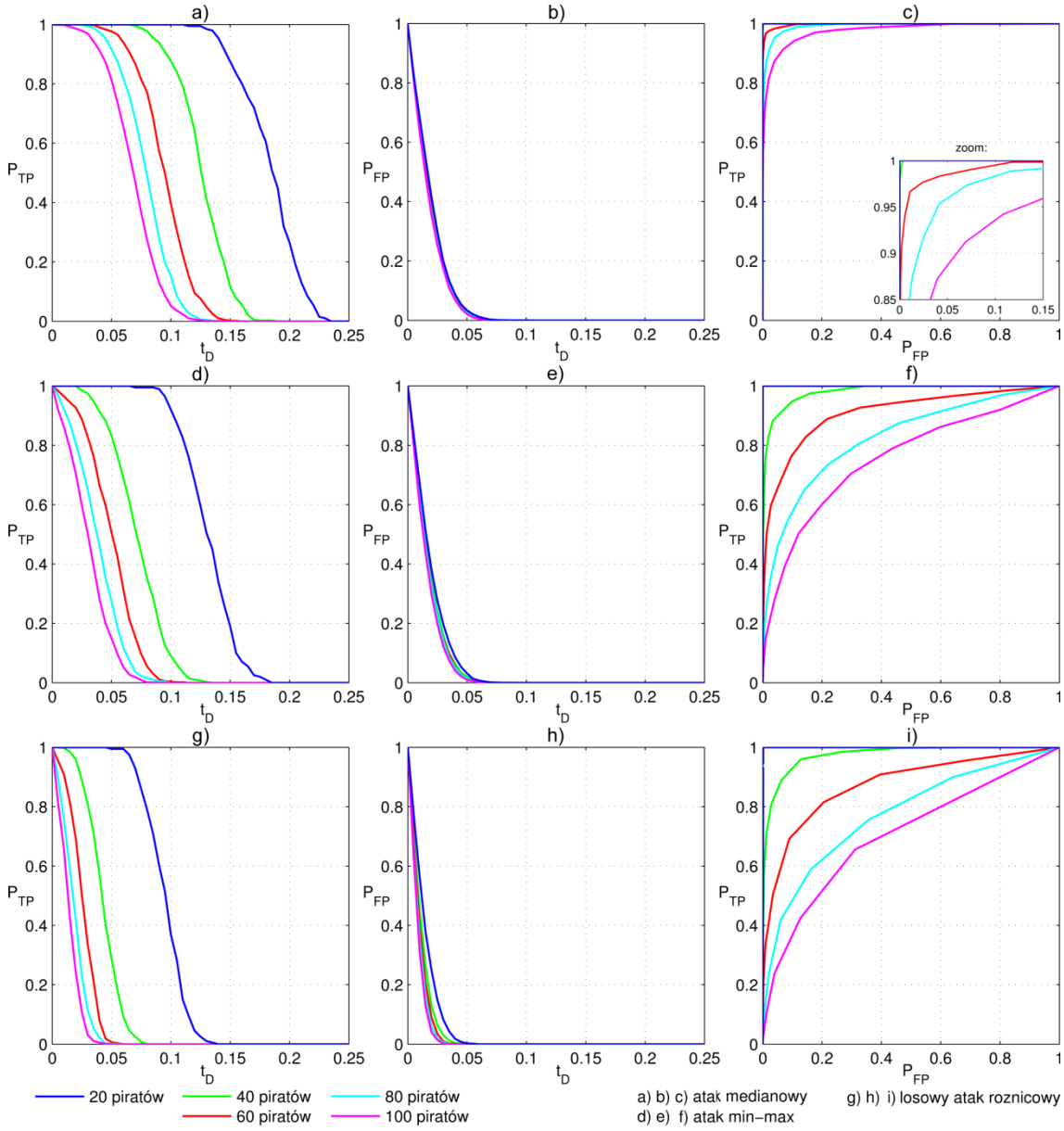


Rys. 9.22. Metoda B: odporność na ataki zmowy (atak medianowy, atak min-max, losowy atak różnicowy) w przypadku zmowy 20, 40, 60, 80 i 100 piratów, opracowanie własne.



Rys. 9.23. Metoda C: odporność na ataki zmowy (atak uśredniający, atak minimum, atak maksimum) w przypadku zmowy 20, 40, 60, 80 i 100 piratów, przy długości serii $Z = 2$, opracowanie własne.

Dla metody C, wyniki badań odporności na atak zmowy metodą uśredniania są następujące. Przy założeniu, że akceptowalne prawdopodobieństwo oskarżenia niewinnego użytkownika wynosi $P_{FP} = 0{,}01$, prawdopodobieństwo identyfikacji pirata wynosi $P_{TP} = 1$ w przypadku zmowy 20, i 40 piratów, $P_{TP} = 0{,}975$ w przypadku 60 piratów, $P_{TP} = 0{,}882$ w przypadku 80 piratów, $P_{TP} = 0{,}687$ w przypadku 100 piratów. Na podstawie powyższych pomiarów i przyjętej definicji odporności można stwierdzić, że metoda C jest odporna na atak zmowy uśredniający realizowany przez 60 lub mniej piratów. Ponadto, w każdej symulacji osiągnięto bezbłędną identyfikację piratów, tj. $P_{TP} = 1$ oraz $P_{FP} = 0$, dla 20 piratów w przypadku ataku minimum, ataku maximum, ataku min-max oraz losowego ataku różnicowego, a nawet dla 40 piratów w przypadku ataku uśredniającego oraz ataku medianowego.



Rys. 9.24. Metoda C: odporność na ataki zmowy (atak medianowy, atak min-max, losowy atak różnicowy) w przypadku zmowy 20, 40, 60, 80 i 100 piratów, przy długości serii $Z = 2$, opracowanie własne.

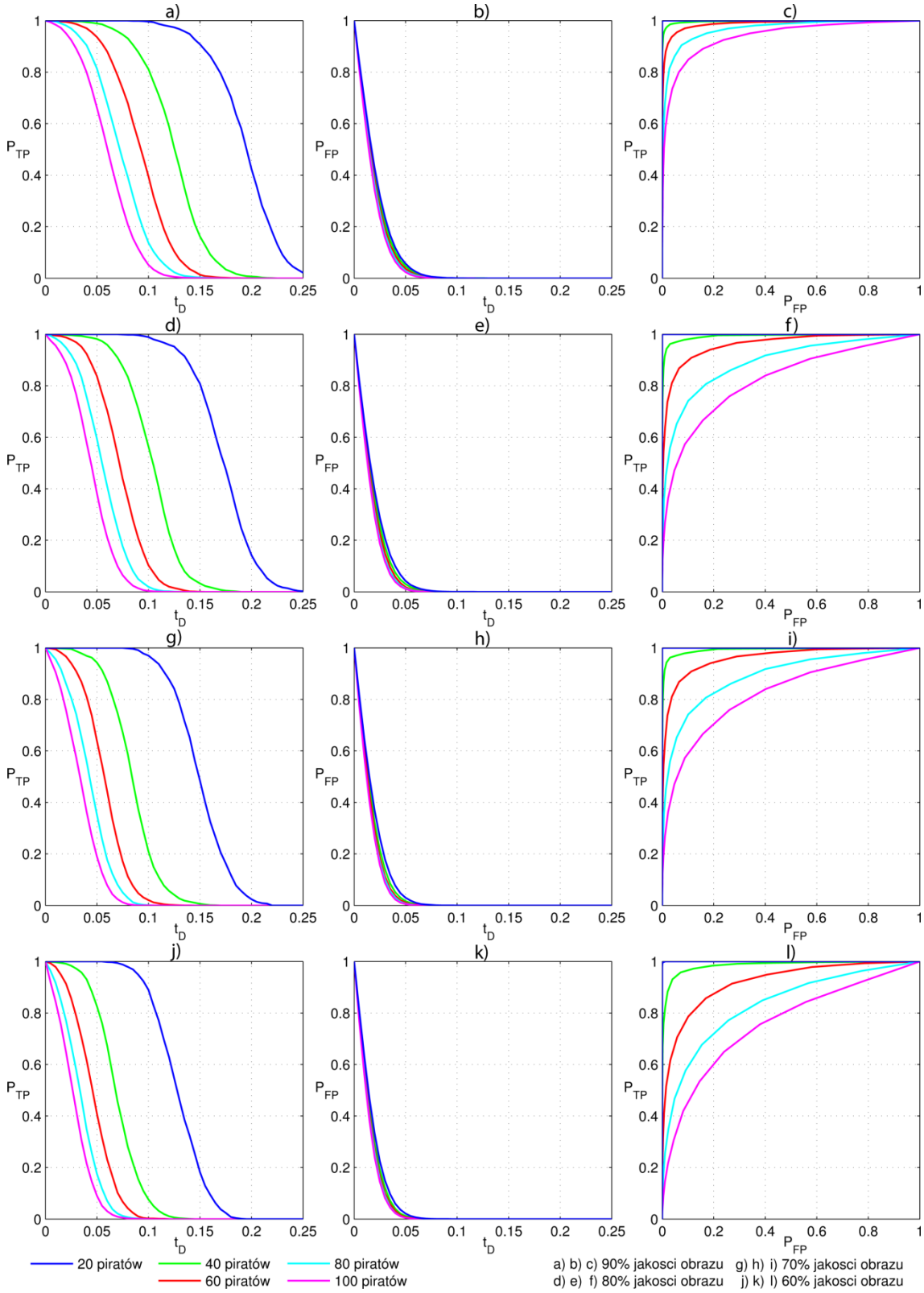
9.6. Odporność na atak zmowy połączony z kompresją

Dla wszystkich zaproponowanych metod JFD, zbadano odporność na ataki zmowy połączone z kompresją. W tych atakach, piracka kopia obrazu jest generowana przez grupę piratów w ataku zmowy metodą uśredniania, a następnie jest kompresowana z wykorzystaniem standardu kompresji JPEG, przy jednoczesnym zachowaniu zadowalającej jakości wizualnej. Dla każdej badanej metody JFD, w każdej z 50 symulacji, zrealizowano 20 różnych ataków zmowy połączonych z kompresją, tzn. atak uśredniający został zrealizowany przez 20, 40, 60, 80 lub 100 piratów, a następnie obraz piracki został skompresowany z zachowaniem 90%, 80%, 70% lub 60% jakości obrazu.

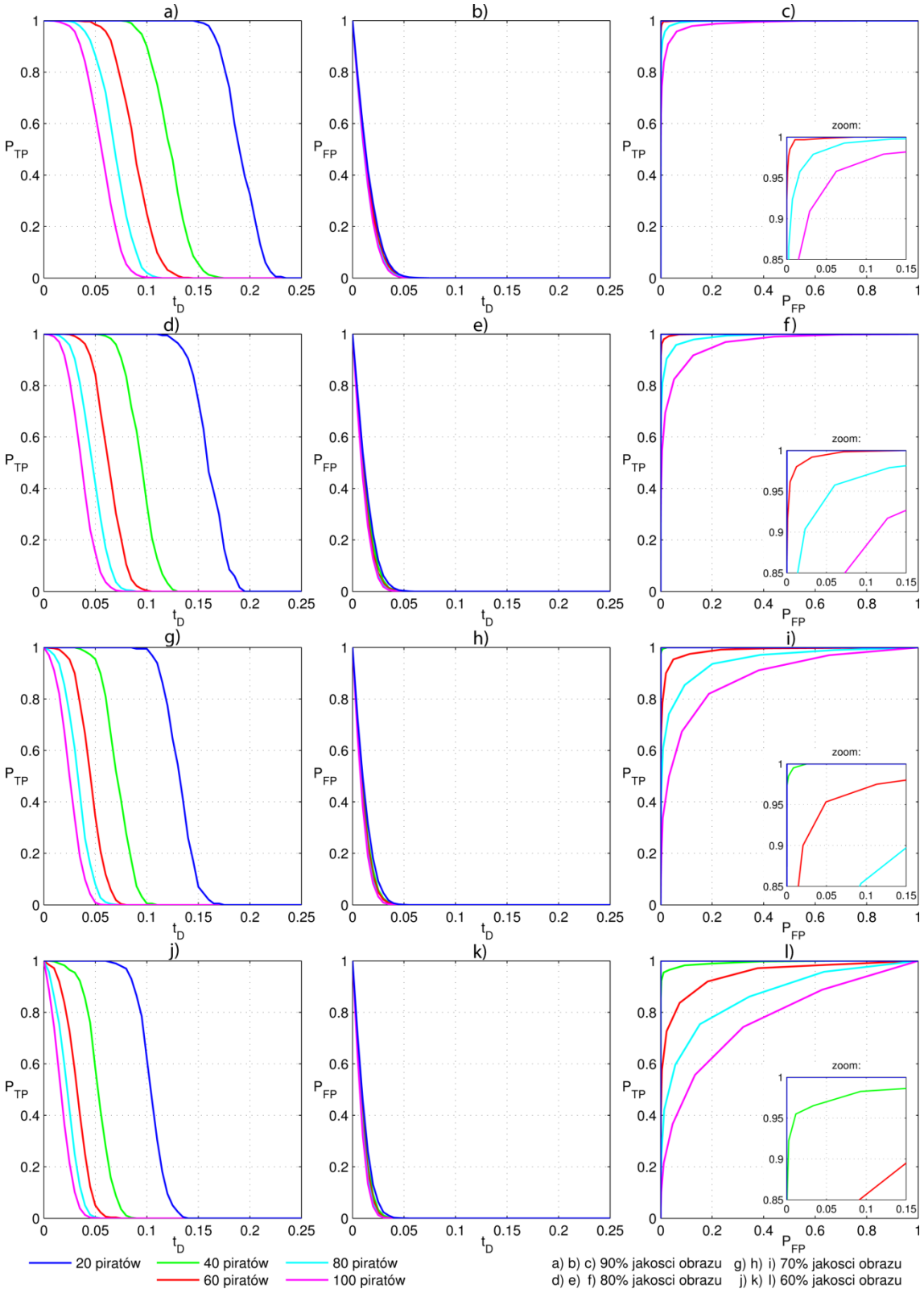
Wyniki badania odporności na ataki zmowy dla metody A przedstawiono na rysunku 9.25., dla metody B na rysunku 9.26., dla metody C na rysunku 9.27. Wykresy w kolejnych wierszach odnoszą się do różnych poziomów kompresji, tj. 90%, 80%, 70% i 60% jakości obrazu. Różne kolory wykresów odnoszą się do różnej liczby piratów w zmowie, tj. 20, 40, 60, 80 i 100 piratów. Wykresy w pierwszej kolumnie przedstawiają zależność P_{TP} od progu detekcji t_D. Na tych wykresach, krzywa najbardziej wysunięta w prawą stronę oznacza większą skuteczność identyfikacji piratów, a dokładniej większą szansę poprawnej identyfikacji pirata. Wykresy w drugiej kolumnie przedstawiają zależność P_{FP} od progu t_D. Na tych wykresach, krzywa bardziej wysunięta w lewą stronę oznacza większą skuteczność identyfikacji piratów, a dokładniej, mniejsze ryzyko oskarżenia niewinnych użytkowników. Należy zwrócić uwagę, że prawdopodobieństwo P_{FP} nie zmienia się znacząco wraz z rosnącą liczbą piratów, co jest pożądaną właściwością. Wykresy w trzeciej kolumnie przedstawiają krzywe ROC, czyli zależność P_{TP} od P_{FP}. Na tych wykresach, krzywa przebiegająca blisko lewego górnego rogu wykresu oznacza bardzo dużą skuteczność, natomiast krzywa bliska przekątnej oznacza bardzo niską skuteczność.

Wyniki badań świadczą o bardzo dużej odporności osadzonych fingerprintów na ataki zmowy połączone z kompresją. Przyjęto taką samą definicję odporności na ataki zmowy, jak w poprzednim podrozdziale. Jeżeli prawdopodobieństwo poprawnie pozytywnej decyzji wynosi $P_{TP} \geq 0{,}9$, przy ustalonym prawdopodobieństwie fałszywie pozytywnej decyzji wynoszącym $P_{FP} = 0{,}01$ oraz przy liczbie piratów *C*, to badana metoda jest odporna na atak zmowy realizowany przez *C* piratów. Natomiast, bezbłędna identyfikacja oznacza, że poprawnie zidentyfikowano wszystkich piratów, bez oskarżania żadnego niewinnego użytkownika.

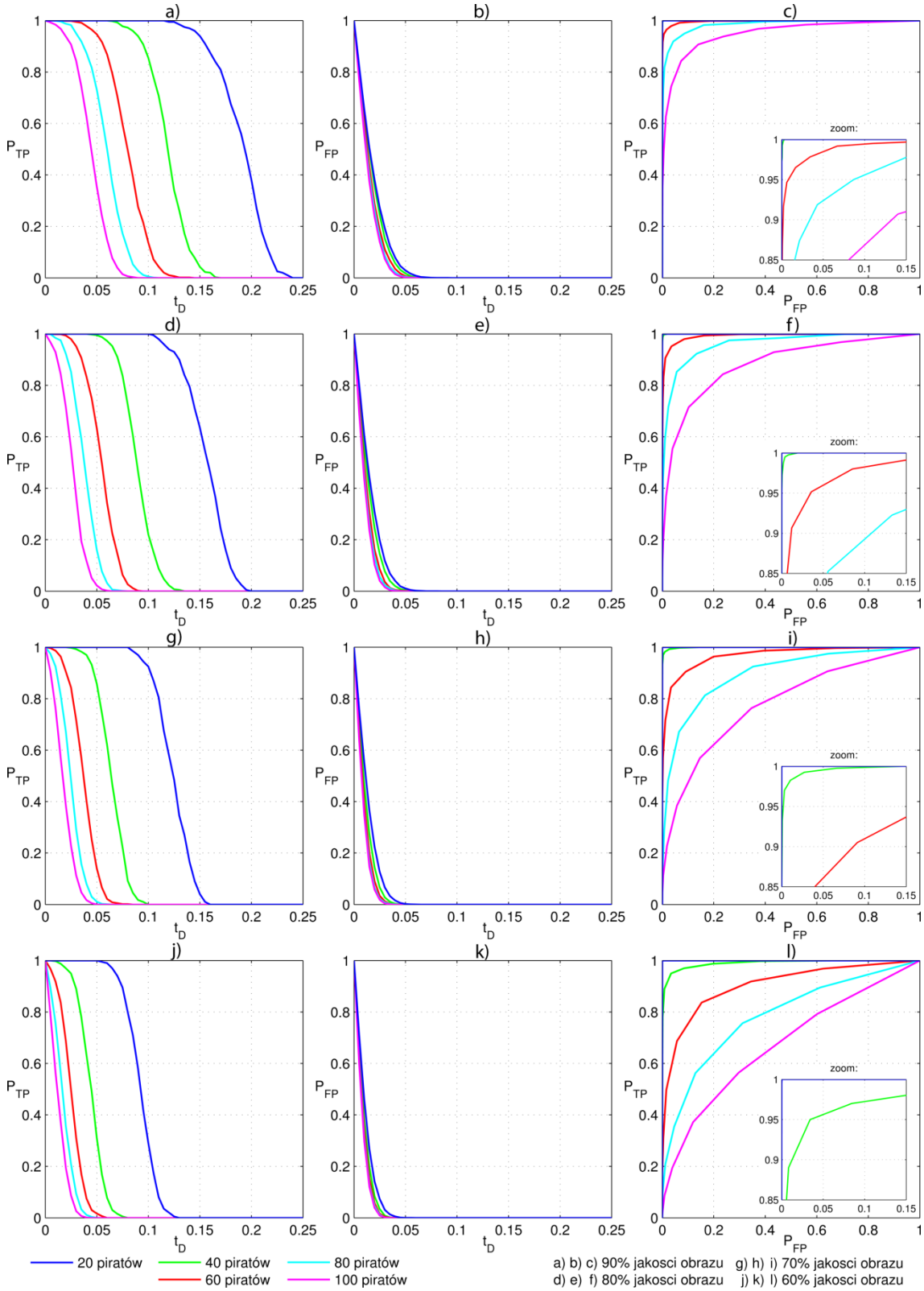
Dla metody A, wyniki badań odporności na atak zmowy z kompresją zachowującą 70% jakości obrazu są następujące. Przy założeniu, że akceptowalne prawdopodobieństwo oskarżenia niewinnego użytkownika wynosi $P_{FP} = 0{,}01$, prawdopodobieństwo identyfikacji pirata wynosi $P_{TP} = 1$ w przypadku zmowy 20 piratów, $P_{TP} = 0{,}916$ w przypadku 40 piratów, $P_{TP} = 0{,}624$ w przypadku 60 piratów, $P_{TP} = 0{,}401$ w przypadku 80 piratów, $P_{TP} = 0{,}258$ w przypadku 100 piratów. Na podstawie powyższych pomiarów i przyjętej definicji odporności można stwierdzić, że metoda A jest odporna na atak zmowy uśredniający połączony z kompresją realizowany przez 40 lub mniej piratów. Ponadto, w każdej symulacji osiągnięto bezbłędną identyfikację piratów, tj. $P_{TP} = 1$ oraz $P_{FP} = 0$, w przypadku ataku zmowy 20 piratów dla wszystkich rozpatrywanych poziomów kompresji.



Rys. 9.25. Metoda A: odporność na atak zmowy o liczbie 20, 40 ,60, 80 i 100 piratów połączony z kompresją zachowującą 90%, 80%, 70% i 60% jakości obrazu, opracowanie własne.



Rys. 9.26. Metoda B: odporność na atak zmowy o liczbie 20, 40 ,60, 80 i 100 piratów połączony z kompresją zachowującą 90%, 80%, 70% i 60% jakości obrazu, opracowanie własne.



Rys. 9.27. Metoda C: odporność na atak zmowy o liczbie 20, 40 ,60, 80 i 100 piratów połączony z kompresją zachowującą 90%, 80%, 70% i 60% jakości obrazu, dla długości serii $Z = 2$, opracowanie własne.

Dla metody B, Wyniki badań odporności na atak zmowy z kompresją zachowującą 70% jakości obrazu są następujące. Przy założeniu, że akceptowalne prawdopodobieństwo oskarżenia niewinnego użytkownika wynosi $P_{FP} = 0{,}01$, prawdopodobieństwo identyfikacji pirata wynosi $P_{TP} = 1$ w przypadku zmowy 20 piratów, $P_{TP} = 0{,}995$ w przypadku 40 piratów, $P_{TP} = 0{,}808$ w przypadku 60 piratów, $P_{TP} = 0{,}612$ w przypadku 80 piratów, $P_{TP} = 0{,}350$ w przypadku 100 piratów. Na podstawie powyższych pomiarów i przyjętej definicji odporności można stwierdzić, że metoda B jest odporna na atak zmowy uśredniający połączony z kompresją realizowany przez 60 lub mniej piratów. Ponadto, w każdej symulacji osiągnięto bezbłędną identyfikację piratów, tj. $P_{TP} = 1$ oraz $P_{FP} = 0$, w przypadku zmowy 20 piratów dla wszystkich rozpatrywanych poziomów kompresji.

Dla metody C, wyniki badań odporności na atak zmowy z kompresją zachowującą 70% jakości obrazu są następujące. Przy założeniu, że akceptowalne prawdopodobieństwo oskarżenia niewinnego użytkownika wynosi $P_{FP} = 0{,}01$, prawdopodobieństwo identyfikacji pirata wynosi $P_{TP} = 1$ w przypadku zmowy 20 piratów, $P_{TP} = 0{,}982$ w przypadku 40 piratów, $P_{TP} = 0{,}681$ w przypadku 60 piratów, $P_{TP} = 0{,}317$ w przypadku 80 piratów, $P_{TP} = 0{,}162$ w przypadku 100 piratów. Na podstawie powyższych pomiarów i przyjętej definicji odporności można stwierdzić, że metoda C jest odporna na atak zmowy uśredniający połączony z kompresją realizowany przez 40 lub mniej piratów. Ponadto, w każdej symulacji osiągnięto bezbłędną identyfikację piratów, tj. $P_{TP} = 1$ oraz $P_{FP} = 0$, w przypadku zmowy 20 piratów dla wszystkich rozpatrywanych poziomów kompresji.

9.7. Porównanie i ocena nowych metod JFD

W tym podrozdziale porównano zaproponowane nowe metody JFD dla obrazów kolorowych ze sobą oraz innymi metodami fingerprintingu znanymi z literatury. Metody te zostały porównane w kontekście niezauważalności osadzonych fingerprintów, odporności na szum, odporności na kompresję, odporności na atak zmowy metodą uśredniania, odporności na ataki zmowy połączone z kompresją, oraz zgodności z koncepcją JFD, która pozwala na wysoką wydajność konstrukcji i dystrybucji, wydajności detekcji oraz trudności implementacji.

Należy zwrócić uwagę, że żadna publikacja znaleziona w literaturze nie zawiera tak szczegółowej i wielokryterialnej analizy danej metody fingerprintingu. W ogólności, publikacje zawierają wyniki badań dotyczące pewnego podzbioru powyższych właściwości. Zazwyczaj badana jest jedynie odporność na kompresję lub atak zmowy. Co więcej, w przypadku ataku zmowy zazwyczaj rozpatrywany jest jedynie liniowy atak zmowy metodą uśredniania. W celu porównania nowych metod z istniejącymi wybrano następujący zbiór metod: Chameleon [10], metoda Kundura i Karthnika [9], metoda z rozpraszaniem widma Cox'a i innych [20], Hillcast [13, 29, 86], fingerprinting w szyfrowaniu kwaternionowym z pracy [37], metoda Parviainena i Parnesa [82], metoda Celika i innych [76], metoda Xu i innych [83], metoda Li i innych [61], hierarchiczny fingerprinting z rozpraszaniem widma [40], oraz metoda oparta na kwantyzacji wektorowej [77, 81]. Przy wyborze powyższych metod sugerowano się możliwością porównania wyników badań przedstawionych w publikacjach z wynikami uzyskanymi dla metod zaproponowanych w niniejszej rozprawie.

Tab. 9.1. Porównanie nowych metod JFD przedstawionych w rozprawie ze sobą oraz z istniejącymi metodami znanymi z literatury.

	Niezauważalność osadzonych fingerprintów	Odporność na szum	Odporność na kompresję	Odporność na ataki zmowy	Odporność na ataki zmowy z kompresją	Zgodność z koncepcją JFD
Metoda A z rozdziału 5	osiągnięta, PSNR ~40dB	osiągnięta, SNR ≥ 5dB	osiągnięta, $Q \geq 10\%$	$C = 60$ (20)	$C = 40$ (20) $Q \geq 70\%$	tak
Metoda B z rozdziału 7	osiągnięta, PSNR ~40dB	osiągnięta, SNR ≥ 5dB	osiągnięta, $Q \geq 10\%$	$C = 100$ (40)	$C = 60$ (20) $Q \geq 70\%$	tak
Metoda C z rozdziału 8	osiągnięta, PSNR ~40dB	osiągnięta, SNR ≥ 5dB	osiągnięta, $Q \geq 10\%$	$C = 60$ (40)	$C = 40$ (20) $Q \geq 70\%$	tak
Chameleon [10]	osiągnięta, osadzanie w bitach LSB	brak badań	brak badań	$C = 3$	brak badań	tak
Kundur i Karthnik [9]	subiektywna, PSNR ~30dB	brak badań	osiągnięta, $Q = 60\%$	brak badań	brak badań	tak
Hillcast [13, 29, 86]	osiągnięta, PSNR ~50dB	brak badań	brak odpor., osadzanie w pikselach	$C = 10$	brak badań	tak
Szyfrowanie kwaternionowe [37]	osiągnięta, PSNR ~55dB	brak badań	brak odpor., osadzanie w pikselach	$C = 30$ (25)	brak badań	nie, chociaż osadzanie po s. nadawczej
Cox i inni [20]	osiągnięta, maskowanie perceptualne	osiągnięta, wynika z założeń	osiągnięta, $Q \geq 5\%$	$C = 5$*	brak badań	nie
Parviainen i Parnes [82]	brak badań	brak badań	brak badań	$C = 10$	brak badań	tak, ale 2 strumienie multicast
Celik i inni [76]	osiągnięta, maskowanie perceptualne	brak badań	brak badań	jak w [20] lub $C = 2$**	brak badań	tak
Xu i inni [83]	osiągnięta, PSNR ~45dB	brak badań	brak badań	$C = 12$	brak badań	tak
Li i inni [61]	osiągnięta, PSNR ~40dB	brak badań	brak badań	$C = 3$*	$C = 3$* $Q \geq 55\%$	nie
Hierarchiczny fingerprinting [40]	osiągnięta, PSNR ~45dB	brak badań	brak badań	$C = 80$	$C = 35$ $Q \geq 75\%$	nie
Kwantyzacja wektorowa [77, 81]	subiektywna, PSNR ~30dB	osiągnięta, zakres szumu 1-16	brak badań	brak badań	brak badań	tak

* W publikacji przedstawiono wyniki dla takiej liczby, badania faktycznego limitu nie zostały przedstawione.

** Możliwe jest wykorzystanie redundancji w tablicach LUT do estymacji obrazu bez fingerprintu.

Porównanie nowych metod JFD przedstawiono w tabeli 9.1. W kolumnach „niezauważalność osadzonych fingerprintów", „odporność na szum" oraz „odporność na kompresję", zamieszczono parametr dla którego osiągnięto w badania daną właściwość. Na przykład, tekst „osiągnięta, $q \geq 10\%$" w kolumnie „odporność na kompresję" oznacza, że dla danej metody odporność na kompresję została osiągnięta przy zastosowaniu kompresji obrazu z zachowaniem 10% lub więcej jakości. W kolumnach „odporność na ataki zmowy" oraz „odporność na ataki zmowy z kompresją" znajdują się maksymalne liczby piratów w zmowie C dla których osiągnięto bardzo duże prawdopodobieństwo poprawnej identyfikacji pirata. Ponadto, jeżeli badania to uwzględniały, w nawiasie podano maksymalną liczbę piratów w zmowie dla których osiągnięto bezbłędną identyfikację piratów. Na przykład, tekst „$C = 60$ (20)" oznacza, że dana metoda jest odporna na ataki zmowy o liczebności 60 piratów z dużym prawdopodobieństwem identyfikacji pirata, a przy zmowie 20 piratów osiągnięto bezbłędną identyfikację wszystkich piratów. Tekst „brak badań" oznacza, ze autorzy danej metody nie przeprowadzili badań danego rodzaju lub nie przedstawili ich wyników w publikacji.

Niezauważalność fingerprintów najczęściej jest mierzona za pomocą szczytowego stosunku sygnału do szumu PSNR, gdzie sygnałem jest obraz oryginalny (sygnał hosta), a szumem osadzony fingerprint. W nielicznych przypadkach, np. w metodzie Cox'a [20] lub metodzie Chameleon [10], niezauważalność fingerprintów jest stwierdzona na podstawie przyjętych założeń. Pomierzone wartości PSNR dla każdej nowej metody A, B i C wynoszą 40 dB lub więcej. Podane wartości PSNR dla metod znanych z literatury zawierają się w stosunkowo szerokim zakresie od 30 dB do 55 dB. Można stwierdzić, że każda nowa metoda charakteryzuje się niezauważalnością osadzonych fingerprintów i jest to poziom niezauważalności porównywalny z metodami znanymi z literatury.

Odporność na szum jest najrzadziej badaną właściwością metod JFD. Wyjątkami są metody, których głównym celem projektowym jest oparcie się wpływowi szumu, np. metody wykorzystujące kwantyzację wektorową [77, 81]. Zbadano, że każda nowa metoda przedstawiona w rozprawie może bezbłędnie identyfikować fingerprint w zaszumionym obrazie przy stosunku SNR od 5 do 30 dB. Posiadanie wyników takich badań pozwala na stwierdzenie odporności na szum dla każdej nowej metody w tej pracy, czego brakuje metodom przedstawionym w literaturze.

Każda z zaproponowanych nowych metod JFD jest odporna na kompresję. Przeprowadzone badania wykazały, że dla każdej z tych metod jest możliwa bezbłędna identyfikacja fingerprintu w skompresowanym obrazie o jakości od 70% do 10%. Podobny poziom odporności na kompresję można zaobserwować u innych metod osadzających fingerprinty w perceptualnie istotnych elementach obrazu oraz w dziedzinie częstotliwości, np. w wybranych współczynnikach DCT, jak np. w metodzie Kundur i Karthnik [9] lub metodzie Cox'a [20]. Istotny jest fakt, że proponowane metody A, B i C charakteryzują się znacznie większą odpornością na kompresję niż metody realizujące osadzanie fingerprintów w dziedzieni przestrzennej, tj. w wartościach pikseli, jak np. metoda Hillcast [13, 29, 86] lub fingerprinting w szyfrowaniu kwaternionowym w pracy [37].

W kontekście odporności na ataki zmowy, jak zostało wcześniej napisane, badania metod fingerprintingu zazwyczaj uwzględniają jedynie atak zmowy metodą uśredniania. Poza nowymi metodami A, B i C, których odporność została zbadana wobec szerokiego zakresu liniowych i nieliniowych ataków zmowy, jednym z takich nielicznych przypadków była me-

toda Li i innych [61], której badania również uwzględniały atak uśredniający, atak maksimum, atak minimum, atak medianowy, atak minmax i losowy atak różnicowy. Wyniki badań nowych metod pokazały, że metoda A jest odporna na atak zmowy realizowany przez 60 piratów, metoda B jest odporna na atak 100 piratów, a metoda C na atak zmowy 60 piratów. Dodatkowo, osiągnięto bezbłędną identyfikację w przypadku zmowy 20 piratów dla metody A i 40 piratów dla metody B i C. Na podstawie tych wyników można stwierdzić, że spośród trzech zaproponowanych nowych metod JFD, metoda B jest najbardziej odporna na ataki zmowy. Ponadto, zaproponowane metody A, B, i C charakteryzują się znacznie większą odpornością na ataki zmowy niż większość metod w literaturze, dla których przeprowdzono analogiczne badań. Jedyną z porównywanych metod, dla której osiągnięto podobny wynik, tj. odporność na atak zmowy 80 piratów, jest metoda hierarchicznego fingerprintingu z rozpraszaniem widma [40].

W kontekście odporności na ataki zmowy połączone z kompresją, wyniki badań nowych metod pokazały, że metoda A jest odporna na atak zmowy realizowany przez 40 piratów, metoda B jest odporna na atak 60 piratów, a metoda C na atak 40 piratów, przy połączeniu zmowy z kompresją z zachowaniem 70% jakości obrazu. Dodatkowo, osiągnięto bezbłędną identyfikację w przypadku zmowy 20 piratów w przypadku wszystkich trzech metod. Na podstawie tych wyników można stwierdzić, że spośród trzech zaproponowanych nowych metod JFD, metoda B jest najbardziej odporna na ataki zmowy połączone z kompresją. Ponadto, zaproponowane metody A, B, i C charakteryzują się znacznie większą odpornością niż metoda Li i innych [61], dla której rozpatrzono zmowę jedynie 3 piratów. Jedyną z porównywanych metod, dla której osiągnięto podobny wynik jest metoda hierarchicznego fingerprintingu z rozpraszaniem widma [40], która jest odporna na atak zmowy 35 piratów przy stosowaniu kompresji z zachowaniem 75% jakości obrazu.

Zgodność z koncepcją JFD, opisana w rozdziale 4., pozwala na osiągnięcie wydajności kontrukcji i dystrybucji. Wydajność konstrukcji dotyczy złożoności obliczeniowej procesu generowania oznakowanych kopii multimediów. Jeżeli złożoność obliczeniowa jest wysoka i szybko rośnie wraz z liczbą klientów, to oznacza, że wydajność konstrukcji oznakowanych danych jest niska. W metodach JFD, złożoność konstrukcji oznakowanej kopii jest minimalna, gdyż jest realizowana w trakcie deszyfrowania, bez konieczności dodatkowych operacji, oraz nie jest zależna od liczby klientów, gdyż jest realizowana po stronie odbiorczej, tj. u każdego klienta niezależnie. Wydajność dystrybucji dotyczy szerokości pasma, które jest używane podczas transmisji wszystkich oznakowanych kopii multimediów przez sieć. Im więcej przepustowości wymaga transmisja, tym niższa jest wydajność dystrybucji oznakowanych danych. W metodach JFD, wymagana przepustowość jest minimalna, dzięki wykorzystaniu transmisji typu multicast. Zaproponowane metody A, B i C w pełni wpisują się w koncepcję łącznego fingerprintingu i deszyfracji, dzięki czemu charakteryzują się większą wydajnością kontrukcji i dystrybucji niż metody, które nie są zgodne z tą koncepcją. Ponadto, metody A, B i C mają również większą wydajność dystrybucji niż metoda JFD Parviainena i Parnesa [82], która wymaga 2 strumieni multicast.

Trzy zaproponowane metody JFD można dodatkowo porównać pod względem wydajności detekcji, czyli złożoności obliczeniowej procesu wydobycia fingerprintu z oznakowanych danych. Porównując równania na właściwą postać fingerprintu w oznakowanej kopii, tj. równość (5.12) dla metody A, równość (7.17) dla metody B, i równość (8.24) dla metody C

można zauważyć, że wydobycie fingerprintu w metodzie B i C jest bardziej złożone obliczeniowo. W metodzie B i C, do obliczenia właściwej postaci osadzonego fingerprintu danego użytkownika wykowane są pewne dodatkowe obliczenia w celu odtworzenia wymaganych kwaternionów. Trudność ekstrakcji fingerprintów w metodzie B i C wynika z nieliniowej zależności pomiędzy właściwą postacią osadzonego fingerprintu, a danymi hosta i kluczem szyfrujących, która została opisana w załączniku A i B. Tymczasem, w metodzie A można obliczyć właściwą postać fingerprintu danego użytkownika poprzez prosty iloczyn macierzy obrazu i macierzy fingerprintu. Z tego względu można stwierdzić, że wydajność detekcji w metodzie A jest większa niż w metodzie B i C. Należy też zwrócić uwagę, iż operacja wydobycia fingerprintu z pirackiej kopii, w każdej z trzech proponowanych metod, jest bardziej złożona niż w dowolnej metodzie osadzającej fingerprint w sposób addytywny. Natomiast, złożoność ekstrakcji fingerprintu ma wpływ na wydajność detekcji. Wobec tego, wydajność detekcji proponowanych metod jest gorsza niż w dowolnej metodzie osadzającej fingerprint w sposób addytywny. Jednakże, wydajność konstrukcji i dystrybucji są znacznie istotniejsze niż wydajność detekcji, gdyż czas trwania detekcji piratów, czyli ekstrakcji fingerprintu oraz następującej po niej identyfikacji piratów, zazwyczaj nie musi być bardzo krótki.

Poza kryteriami przedstawionymi w tabeli 9.1., trzy zaproponowane metody JFD można porównać w kontekście trudności implementacji. Metoda A wykorzystuje stosunkowo proste operacje algebraiczne, które są zaimplementowane w większości popularnych języków programowania lub można niskim nakładem pracy oprogramować funkcje do ich realizacji. Natomiast, metody B i C wykorzystują operacje algebry kwaternionów, które bardzo rzadko są zaimplementowane w popularnych językach. W takich przypadkach konieczne jest wykorzystanie dodatkowej biblioteki, która udostępnia odpowiednie narzędzia do wykonywania operacji rachunku kwaternionowego lub stworzenie własnej klasy, która zastąpiłaby taką bibliotekę. W tym kontekście, metody B i C wykorzystujące rachunek kwaternionowy są trudniejsze w implementacji. Do prowadzenia badań nad metodami zaproponowanymi w tej rozprawie wykorzystano zestaw narzędzi kwaternionowych [99] dla programu Matlab.

9.8. Podsumowanie

Przeprowadzono obszerne badania wszystkich trzech proponowanych metod JFD, w których zbadano niezauważalność osadzonych fingerprintów, odporność na szum, odporność na kompresję, odporność na ataki zmowy oraz odporność na ataki zmowy połączone z kompresją. W przeprowadzonych symulacjach osiągnięto niezauważalność osadzonych fingerprintów, dla każdej badanej metody. W przypadku ataków szumem oraz ataków kompresją osiągnięto bezbłędną detekcję piratów, dla każdej badanej metody. W przypadku liniowych ataków zmowy, na podstawie otrzymanych wyników badań i przyjętych definicji określono, że proponowane metody pozwalają na bardzo skuteczną identyfikację piratów, tj. $P_{TP} \geq 0{,}9$, oraz $P_{FP} = 0{,}01$, jeżeli liczba piratów w zmowie wynosi nie więcej niż 60 dla metody A i C, i 100 dla metody B. Ponadto, dla metody A, osiągnięto bezbłędną identyfikację piratów, tj. $P_{TP} = 1$, oraz $P_{FP} = 0$, dla 20 piratów w zmowie we wszystkich rozpatrywanych typach ataków zmowy z wyjątkiem losowego ataku różnicowego Natomiast, dla metody B i C, osiągnięto bezbłędną identyfikację dla 40 piratów w ataku uśredniającym i ataku medianowym,

a także dla 20 piratów we wszystkich pozostałych rozpatrywanych atakach zmowy. W przypadku ataków zmowy połączonych z kompresją określono, że proponowane metody pozwalają na bardzo skuteczną identyfikację piratów, tj. $P_{TP} \geq 0{,}9$, oraz $P_{FP} = 0{,}01$, jeżeli liczba piratów w zmowie wynosi nie więcej niż 40 dla metody A i C, i 60 dla metody B. Ponadto, osiągnięto bezbłędną identyfikację, tj. $P_{TP} = 1$, oraz $P_{FP} = 0$, dla 20 piratów w zmowie dla wszystkich rozpatrywanych poziomów kompresji, dla każdej badanej metody. Pomimo, że podczas badań znakowane były obrazy nieruchome, to wyniki badań można odnieść również do danych wideo, gdyż w analogiczny sposób mogą być znakowane wybrane ramki w sekwencjach wideo, np. ramki typu I w kodekach MPEG.

Pomierzone właściwości proponowanych metod JFD zostały porównane z właściwościami innych opublikowanych rozwiązań. Nowe metody JFD dla obrazów kolorowych zostały ocenione pod względem niezauważalności osadzanych fingerprintów, odporności na szum, odporności na kompresję, odporności na ataki zmowy, odporności na ataki zmowy z kompresją, wydajności konstrukcji, wydajności dystrybucji oraz wydajności detekcji. W wyniku porównania metod A, B i C z metodami znanymi z literatury, stwierdzono przewagę nowych metod w kontekście odporności na wyżej wymienione zagrożenia. Ponadto, zaletą metod kwaternionowych jest to, że wykorzystują łącznie wszystkie trzy kanały kolorów, co można określić mianem fingerprintingu całościowego (holistycznego). Jednocześnie, wskazano ograniczenia nowych metod w postaci konieczności implementacji operacji kwaternionowych oraz wydajności detekcji niższej niż metody fingerprintingu addytywnego. Na podstawie uzyskanych wyników badań, przewiduje się, że nowe metody JFD mogą przynieść znacznie większe korzyści niż inne rozwiązania, w postaci skuteczniejszej identyfikacji piratów, przy ekonomicznej dystrybucji multimediów w usługach wideo na żądanie lub strumieniowania wideo.

10. SPRAWIEDLIWY PROTOKÓŁ DYSTRYBUCJI DLA METOD JFD

Metody fingerprintingu są projektowane z myślą o ochronie usługodawców przed działaniami nieuczciwych klientów, podczas gdy uczciwi klienci pozostają narażeni na działania nieuczciwych usługodawców. W rezultacie, metody te nie są sprawiedliwe z punktu widzenia klienta. Oczywiście, celem metod fingerprintingu jest ochrona praw autorskich poprzez śledzenie piratów (ang. *traitor tracing*), czyli ochrona interesów strony dystrybucyjnej, jednakże możliwe jest wprowadzenie pewnych dodatkowych mechanizmów, które zapewniłyby sprawiedliwą ochronę również stronie odbiorczej. Problem braku ochrony klienta również dotyczy metod łącznego fingerprintingu i deszyfracji, pomimo ich wyjątkowej cechy, jaką jest osadzanie fingerprintów po stronie odbiorczej. W tym rozdziale wyjaśniono, na czym polega problem braku ochrony klienta oraz zaproponowano rozwiązanie w postaci sprawiedliwego protokołu zarządzania dystrybucją dla metod JFD.Proponowany protokół jest wzorowany na rozwiązaniach znanych z literatury, ale różni się tym, że istniejące protokoły zarządzania dystrybucją zakładają addytywny fingerprinting realizowany po stronie nadawczej, natomiast proponowany protokół jest dostosowany do koncepcji łącznego fingerprintingu i deszyfracji i może być używany razem z dowolną metodą JFD.

10.1. Problem braku ochrony klienta

Problem braku ochrony klienta składa się z dwóch zagadnień, które w literaturze zostały określone jako problem praw klienta (ang. *customer's rights problem*) oraz problem niepowiązania (ang. *unbinding problem*) [100, 101, 102, 103]. W ogólności, oba problemy opisują sytuację, w której sprzedawca dysponuje środkami umożliwiającymi wykorzystanie fingerprintingu do działania na szkodę klienta. Jako sprzedawcę należy rozumieć stronę dystrybucyjną metody fingerprintingu, np. usługodawcę VoD, a klientem jest strona odbiorcza, np. użytkownik usługi VoD.

Problem praw klienta (ang. *customer's rights problem*) pojawia się, gdy nieuczciwy sprzedawca może upozorować piractwo w celu wrobienia uczciwego klienta. Problem ten dotyczy metody fingerprintingu, w której sprzedawca ma dostęp do każdej oznakowanej kopii dystrybuowanych multimediów lub posiada zbiór danych wystarczający do odtworzenia każdej oznakowanej kopii multimediów, w takiej postaci, w jakiej otrzymują je klienci. Problem ten wynika z błędnego założenia, że sprzedawca zawsze jest godny zaufania. Należy zwrócić uwagę, że nawet w przypadku uczciwego sprzedawcy możliwe jest, że oznakowana kopia multimediów wycieknie z powodu słabego punktu w systemie sprzedaży i błędnie wskaże, że uczciwy klient dopuścił się piractwa. Problem praw klienta istnieje, jeżeli sprzedawca jest jedyną stroną odpowiedzialną za generowanie i osadzanie fingerprintów w dystrybuowanych multimediach oraz nie istnieje protokół wprowadzający odpowiednie mechanizmy ustalania prawdziwego źródła pirackiej kopii.

W klasycznym modelu fingerprintingu, proces osadzania cyfrowych odcisków palca jest w pełni kontrolowany przez sprzedawcę. Sprzedawca jest w stanie, świadomie lub nieświadomie, utworzyć kolejną kopię oznakowaną fingerprintem, który został już przypisany do pewnego klienta, oraz wysłać ją do strony trzeciej. Później, sprzedawca może spróbować pozwać tego uczciwego klienta wrabiając go w udostępnienie multimediów stronie trzeciej.

Z drugiej strony, nieuczciwy klient może nielegalnie rozpowszechniać swoją oznakowaną kopię oraz twierdzić, że nielegalne kopie pochodzą od sprzedawcy, gdyż ten posiada wystarczający zbiór danych, aby takie kopie wygenerować i rozpowszechniać. Oznacza to, że klient, którego fingerprint został znaleziony w pirackiej kopii, paradoksalnie nie może być uznawany za pirata, gdyż nie istnieją dowody świadczące na korzyść sprzedającego.

Problem praw klienta jest bardzo wyraźny w klasycznym modelu fingerprintingu, w którym fingerprint jest generowany i osadzany w danych autonomicznie przez stronę dystrybucyjną, bez żadnego nadzoru. Jednakże, problem ten dotyczy również metod JFD, gdyż strona dystrybucyjna ma dostęp do zaszyfrowanych danych multimedialnych oraz do unikatowych kluczy deszyfrujących użytkowników. Wówczas, sprzedawca może przeprowadzić deszyfrowanie z użyciem jednego z tych kluczy deszyfrujących i uzyskać oznakowaną kopię danych, przeznaczoną dla danego użytkownika.

Problem niepowiązania (ang. *unbinding problem*) pojawia się, gdy nieuczciwy sprzedawca może przenieść fingerprint z pirackiej kopii do innego obrazu o wyższej cenie, w celu upozorowania piractwa. Problem ten dotyczy metody fingerprintingu, w której nie istnieją mechanizmy wiążące wybrany fingerprint z dystrybuowanymi danymi lub konkretną transakcją. Wówczas, sprzedawca, który odkrył istniejącą piracką kopię może przenieść fingerprint osadzony w pirackiej kopii do innej treści multimedialnej o wyższej cenie, aby upozorować piractwo, które przyniosłoby mu większą rekompensatę, w przypadku skazania pirata.

W celu pełnej ochrony klienta, poza rozwiązaniem problemu praw klienta oraz problemu niepowiązania, konieczne jest, aby protokół umożliwiał zapewnienie wzajemnego uwierzytelnienia. Wzajemne uwierzytelnienie jest niezbędne, aby każda ze stron ustaliła określony poziom zaufania wobec drugiej, zwłaszcza w przypadku, gdy sprzedawca i klient są znacznie oddaleni geograficznie. Sprzedawcy muszą znać fizyczną tożsamość klientów, aby w przypadku naruszenia praw autorskich mogli ubiegać się w sądzie o wymierzenie sprawiedliwości nieuczciwym użytkownikom. Z drugiej strony, klienci powinni mieć możliwość zachowania anonimowości, jeżeli wymagałyby tego okoliczności. Wówczas, fizyczna tożsamość klienta powinna zostać ujawniona dopiero wtedy, kiedy udowodnione zostało, że jest on winny piractwa. Jednocześnie, klienci powinni znać tożsamość sprzedawcy i mieć pewność, że sprzedawca ma prawo do posiadania i rozpowszechniania zamawianych danych multimedialnych. Oznacza to, że przy transakcji sprzedawca musi przedstawić dowód upoważniający go do posiadania i rozpowszechniania tych danych.

10.2. Proponowane rozwiązanie

W literaturze przedstawiono rozwiązania powyższych problemów w postaci sprawiedliwych protokołów zarządzania transakcją i generowania fingerprintów [100, 101, 102, 103]. Jednakże, rozwiązania te realizują osadzanie fingerprintów po stronie nadawczej, zgodne z klasycznym modelem fingerprintingu, lub zakładają addytywny sposób osadzania fingerprintów w danych multimedialnych. Niestety, w metodach JFD fingerprinty nigdy nie są osadzane po stronie nadawczej, a w wielu przypadkach również nie są osadzane w sposób addytywny. Ponadto, rozwiązania te wykorzystują szyfrowanie homomorficzne, czyli szyfrowanie, które pozwala na operowanie na zaszyfrowanych danych bez ich deszyfrowania, w celu osadzenia fingerprintu w zaszyfrowanym obrazie bez zdradzania go stronie nadawczej [100, 101,

102, 103]. W przypadku metod JFD, w analogiczny sposób należałoby osadzić fingerprint w kluczu deszyfrującym, ale w większości przypadków nie jest to możliwe ze względu na konstrukcję klucza deszyfrującego, charakterystyczną dla danej metody. W rezultacie, istniejące protokoły nie mogą być zastosowane do nowych metod JFD zaproponowanych w niniejszej pracy i istnieje potrzeba zaprojektowania nowego, ogólniejszego rozwiązania.

Proponowane rozwiązanie jest rozwinięciem sprawiedliwych protokołów zarządzania dystrybucją dla fingerprintingu [100, 101, 102, 103], które jest dostosowane do ogólnych założeń koncepcji łącznego fingerprintingu i deszyfracji. Należy zwrócić uwagę, że protokół przedstawiony w tym rozdziale w żaden sposób samodzielnie nie realizuje cyfrowego znakowania danych, ale może być wykorzystany do współpracy z dowolną metodą łącznego fingerprintingu i deszyfracji, np. z nowymi metodami przedstawionymi w tej pracy. Przeznaczeniem proponowanego protokołu jest osiągnięcie następujących celów: rozwiązanie problemu praw klienta (ang. *customer's right problem*), rozwiązanie problemu niepowiązania (ang. *unbinding problem*), zapewnienie wzajemnego uwierzytelnienia stron, oraz opcjonalne zapewnienie anonimowości klienta. Zakłada się, że współpracująca metoda JFD zapewnia ochronę praw autorskich poprzez tropienie piratów (ang. *traitor tracing*) oraz wizualne bezpieczeństwo przesyłanych obrazów poprzez szyfrowanie selektywne (ang. *partial encryption*).

W rozpatrywanym modelu wymiany informacji występuje 5 następujących ról:

- Sprzedawca S – strona dystrybucyjna, która chce osiągnąć korzyść majątkową z tytułu dystrybucji treści multimedialnych. Sprzedawca może być właścicielem treści multimedialnych lub uprawnionym dystrybutorem, ale w każdym przypadku oczekuje ochrony przed działaniami nieuczciwego klienta.
- Klient K – strona odbiorcza, która chce nabyć kopię treści multimedialnych. Klient oczekuje ochrony przed działaniami nieuczciwego sprzedawcy.
- Zaufany urząd certyfikacji CA (ang. *Certificate Authority*) – odpowiedzialny za wydawanie anonimowych certyfikatów. CA poświadcza prawdziwość pseudonimów klientów oraz ujawnia ich prawdziwą tożsamość w przypadku udowodnienia winy.
- Centrum generowania kluczy KGC (ang. *Key Generation Center*) – zaufana trzecia strona odpowiedzialna za generowanie kluczy deszyfrujących dla wykorzystywanej metody JFD, na podstawie danych od sprzedawcy i klienta. KGC zapewnia, że podczas dystrybucji multimediów fingerprinty oraz klucze deszyfrujące nie są znane sprzedawcy.
- Centrum identyfikacji piratów PIC (ang. *Pirate Identification Center*) – zaufana trzecia strona odpowiedzialna za wydobycie fingerprintu z pirackiej kopii, identyfikację piratów i przekazanie tożsamości piratów sprzedawcy. PIC zapewnia, że podczas procesu detekcji piratów fingerprinty i klucze deszyfrujące nie są znane sprzedawcy.

PIC oraz KGC łącznie stanowią zaufaną trzecią stronę, ale w poniższym opisie będą rozpatrywane oddzielnie ze względu na ich rozłączną funkcjonalność.

Protokół wykorzystuje infrastrukturę klucza publicznego PKI (ang. *Public Key Infrastructure*). Zakłada się, że infrastruktura PKI została wdrożona oraz, że każda ze stron posiada parę kluczy asymetrycznych, czyli klucz publiczny i klucz tajny, a także certyfikat wydany

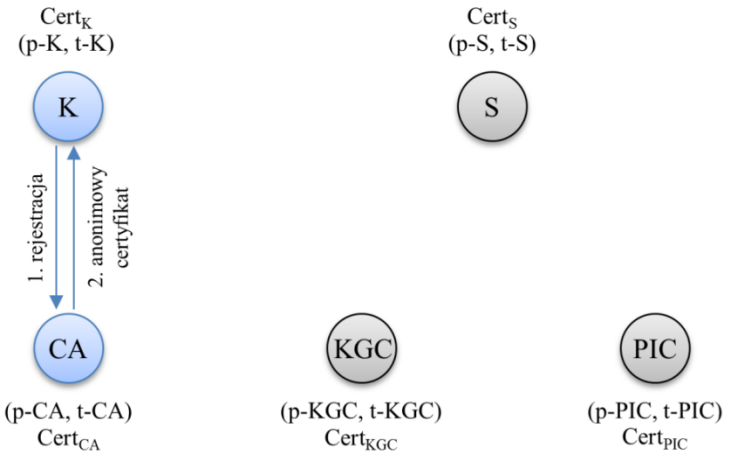
przez zaufany urząd certyfikacji. Certyfikaty zawierają: klucz publiczny, tożsamość posiadacza certyfikatu, informację o upoważnionej jednostce podpisującej certyfikat, informację o użytym algorytmie podpisu cyfrowego, oraz okres ważności certyfikatu. Zakłada się również, że podpisy urzędów ceryfikacji CA mogą być zweryfikowane przez każdą stronę, ponieważ znane są klucze publiczne CA lub możliwe jest użycie łańcucha zależności certyfikatów w celu ich określenia. Ponadto, sprzedawca S ma dostęp do certyfikatów PIC i KGC i uznaje je za zaufane. Sprzedawca posiada również cyfrowy dowód posiadania prawa do dystrybucji rozpatrywanych treści multimedialnych.

Działanie proponowanego protokołu dla metod JFD składa się z trzech etapów: rejestracji anonimowych klientów, dystrybucji kluczy deszyfrujących oraz identyfikacji piratów. W opisie wszystkich etapów protokołu wykorzystano następujące oznaczenia:

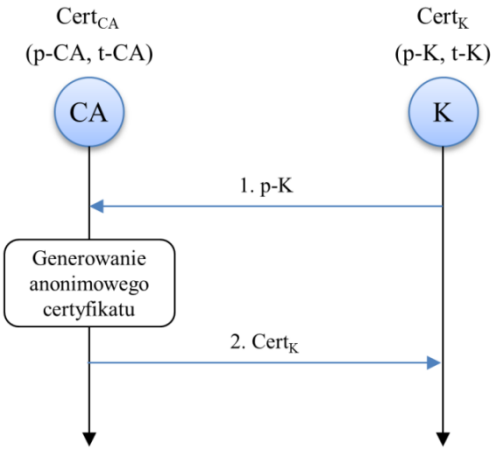
- (p-A, t-A) – para kluczy asymetrycznych strony A, gdzie p-A jest kluczem publicznym strony A, a t-A jest kluczem tajnym strony A,
- Cert_A – certyfikat strony A wydany i podpisany przez zaufany urząd certyfikacji, którego podpis jest możliwy do zweryfikowania,
- $\mathrm{Sign}_A(arg)$ – dane *arg* z podpisem cyfrowym strony A,
- $\mathrm{Encr}_{p\text{-}A}(arg)$ – dane *arg* zaszyfrowane kluczem publicznym strony A, możliwy do odszyfrowania tylko kluczem tajnym strony A,
- *ord* – zamówienie usługi, nakłada prawa i zobowiązania na sprzedawcę i klienta, zgodnie z regulaminem usługi oraz określa zamawianą treść multimedialną,
- *ack* – potwierdzenie przyjęcia zamówienia usługi,
- *prf* – cyfrowy dowód posiadania prawa do dystrybucji pożądanych multimediów,
- *res* – wyniki identyfikacji piratów wykazujące winę nieuczciwych klientów,
- *pir* – tożsamość klientów, których winę wykazano podczas identyfikacji piratów,
- X – obraz oryginalny,
- Y – obraz zaszyfrowany zgodnie z wykorzystywaną metodą JFD,
- E – klucz szyfrujący metody JFD,
- D – klucz deszyfrujący metody JFD,
- F_K – składowa fingerprintu klienta pochodząca od klienta,
- F_S – składowa fingerprintu klienta pochodząca od sprzedawcy,
- F_{ord} – składowa fingerprintu klienta związana z zamówieniem,
- F – fingerprint klienta,
- α – współczynnik siły osadzania fingerprintu,
- ψ_1 – funkcja generująca F_{ord} na podstawie *org*,
- ψ_2 – funkcja generująca F na podstawie F_K, F_S oraz F_{ord}
- ψ_3 – funkcja generująca D na podstawie E, F oraz α zgodnie z metodą JFD,
- *opt:arg* – oznaczenie, że argument *arg* jest opcjonalny i występuje tylko wtedy, kiedy wykorzystywana metoda JFD tego wymaga,
- $\{arg\}$ – zbiór argumentów *arg* dla wszystkich klientów, tzn. zbiór wszystkich *arg* przypisanych do poszczególnych klientów.

Pierwszy etap, czyli rejestracja anonimowych klientów, nie jest obowiązkowy i zachodzi tylko wtedy, kiedy klient wymaga zapewnienia anonimowości. Rejestracja, w celu uzyskania anonimowego certyfikatu, rozpoczyna się od tego, że klient wysyła swój klucz publiczny do CA (rys. 10.2. wiadomość 1). CA odbiera klucz publiczny klienta i generuje ano-

nimowy certyfikat, który jest standardowym cyfrowym certyfikatem, ale w miejscu prawdziwej tożsamości klienta znajduje się jego pseudonim. CA jest odpowiedzialne za powiązanie prawdziwej tożsamości klienta z wydanym anonimowym certyfikatem. Do czasu udowodnienia winy danego klienta, to powiązanie nie może być ujawnione żadnej innej stronie. Następnie, anonimowy certyfikat klienta $Cert_K$, podpisany przez CA, jest wysyłany do klienta (rys. 10.2. wiadomość 2). Przebieg etapu rejestracji klientów został przedstawiony na rysunkach 10.1. oraz 10.2. Alternatywnie, klient może całkowicie ominąć ten etap i w kolejnym etapie wykorzystywać swój normalny certyfikat, jeżeli nie zależy mu na anonimowości.



Rys. 10.1. Ogólny schemat etapu rejestracji anonimowych klientów, opracowanie własne.



Rys. 10.2. Szczegółowy schemat rejestracji anonimowych klientów, opracowanie własne.

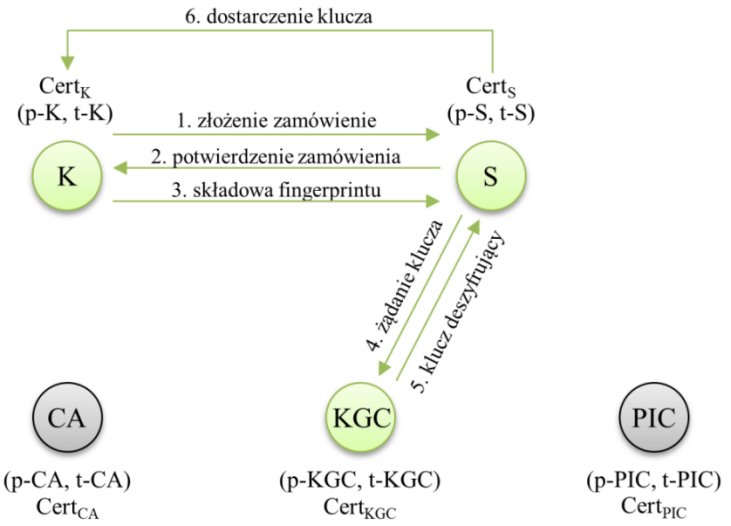
Sprzedawca dysponuje obrazem oryginalnym *X*, który zaszyfrował do postaci *Y* za pomocą klucza *E*, zgodnie z wykorzystywaną metodą JFD. Etap dystrybucji danych i kluczy rozpoczyna się z chwilą, gdy klient składa zamówienie na usługę dystrybucji obrazu *X*. W tym celu klient wysyła do sprzedawcy swój certyfikat $Cert_K$ oraz podpisane zamówienie $Sign_K(ord)$ (rys. 10.4. wiadomość 1). Klient może wykorzystać certyfikat zawierający jego prawdziwą tożsamość albo certyfikat anonimowy, otrzymany w poprzednim etapie. Format zamówienia *ord* został upubliczniony wcześniej przez sprzedawcę, np. na jego stronie internetowej. Sprzedawca weryfikuje podpis w $Sign_K(ord)$ i przyjmuje zamówienie usługi. W tej chwili, sprzedawca zna tożsamość klienta lub jego pseudonim poświadczony przez CA.

Sprzedawca wysyła do klienta swój certyfikat Cert_S, certyfikat KGC Cert_{KGC} oraz podpisane potwierdzenie przyjęcie zamówienia, z dowodem posiadania prawa do dystrybucji $\text{Sign}_S(ack, prf)$ (rys. 10.4. wiadomość 2). Klient weryfikuje podpis w $\text{Sign}_S(ack, prf)$ a następnie, jeżeli wszystkie informacje są poprawne, rozpoczyna się proces generowania fingerprintu klienta i klucza deszyfrującego. Wzajemne uwierzytelnienie stron zostało osiągnięte poprzez wymianę certyfikatów pomiędzy S i K i weryfikację podpisów.

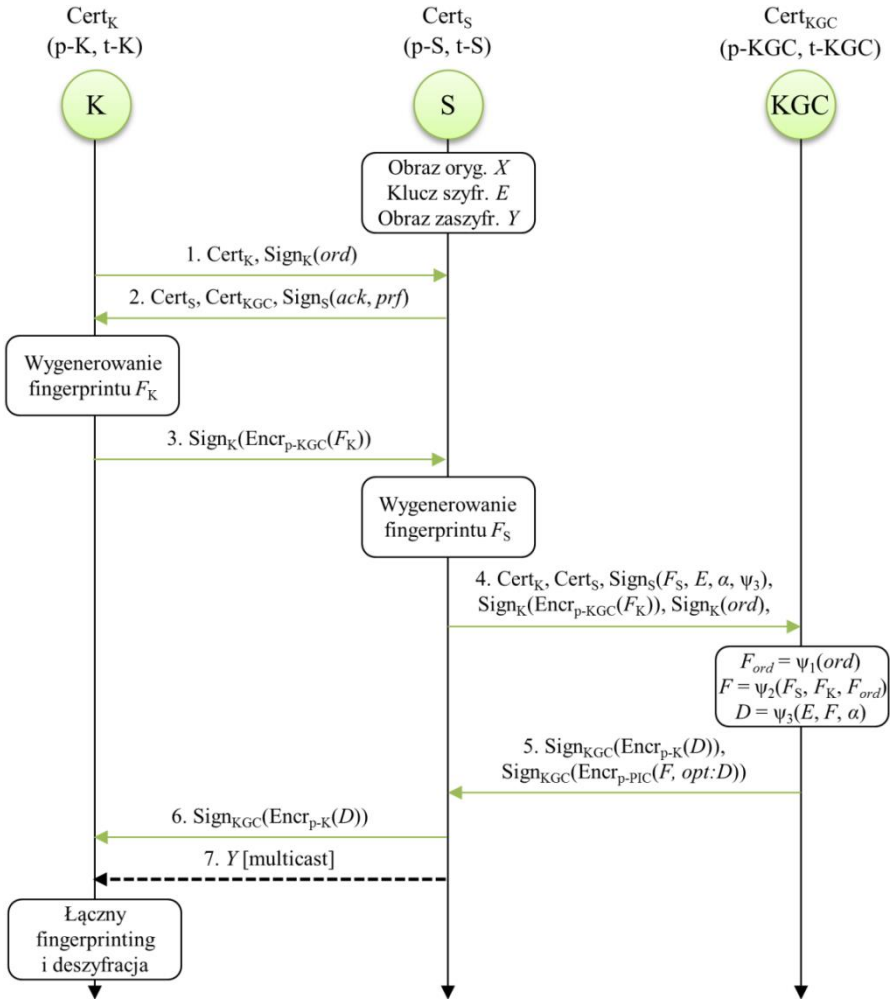
Najpierw, klient generuje swoją składową fingerprintu F_K. Składowa F_K zostaje zaszyfrowana za pomocą klucza publicznego KGC podanego w Cert_{KGC} oraz podpisana przez klienta. Zaszyfrowana i podpisana składowa fingerprintu $\text{Sign}_K(\text{Encr}_{p\text{-}KGC}(F_K))$ zostaje wysłana do sprzedawcy (rys. 10.4. wiadomość 3) z przeznaczeniem do przekazania do KGC. Sprzedawca nie ma dostępu do składowej pochodzącej od klienta, ponieważ nie jest w stanie odszyfrować danych przeznaczonych do KGC, tj. nie posiada klucza tajnego KGC. Następnie, sprzedawca generuje swoją składową fingerprintu F_S i podpisuje ją, wraz z kluczem szyfrującym E, współczynnikiem siły osadzania fingerprintu α, oraz funkcją ψ_3. Funkcja ψ_3 definiuje strukturę klucza deszyfrującego i jest charakterystyczna dla wykorzystywanej metody JFD, np. dla nowej metody macierzowej z rozdz. 5. jest opisana wzorem (5.1), a dla nowej metody kwaternionowej z rozdz. 7. wzorem (7.3). Po podpisaniu swoich danych, sprzedawca wysyła Cert_K, Cert_S, $\text{Sign}_S(F_S, E, \alpha, \psi_3)$, $\text{Sign}_K(\text{Encr}_{p\text{-}KGC}(F_K))$ oraz $\text{Sign}_K(ord)$ do KGC z żądaniem wygenerowania fingerprintu i klucza deszyfrującego dla klienta (rys. 10.4. wiadomość 4).

KGC weryfikuje podpisy obu stron na podstawie przedstawionych certyfikatów oraz odszyfrowuje składową F_K. Po pomyślnej weryfikacji podpisów, KGC generuje trzecią składową fingerprintu F_{ord} za pomocą funkcji ψ_1 na podstawie ord. Funkcja ψ_1 jest znana tylko KGC i może uwzględniać np. identyfikator zamówienia. Fingerprint użytkownika F zostaje utworzony przez KGC za pomocą funkcji ψ_2 na podstawie F_S, F_K i F_{ord}. Funkcja ψ_2 jest znana tylko KGC i może łączyć składowe w dowolny sposób, np. składowe ciągi mogą zostać połączone w jeden, składowe mogą stanowić ziarno dla generatora ciągów pseudolosowych, albo funkcja ψ_2 może wykorzystywać techniki współdzielenia sekretów (ang. *secret sharing*). Ostatecznie, klucz deszyfrujący klienta D zostaje utworzony za pomocą funkcji ψ_3 na podstawie E, F i α. Następnie, KGC szyfruje D za pomocą klucza publicznego klienta podanego w Cert_K oraz szyfruje F za pomocą klucza publicznego PIC. Jeżeli wykorzystywana metoda JFD wymaga znajomości kluczy deszyfrujących użytkowników do identyfikacji piratów, to do zaszyfrowanego F można dodać D. Oba szyfrogramy zostają podpisane przez KGC i wysłane do sprzedawcy (rys. 10.4. wiadomość 5), przy czym $\text{Sign}_{KGC}(\text{Encr}_{p\text{-}K}(D))$ jest przeznaczony do przekazania klientowi, a $\text{Sign}_{KGC}(\text{Encr}_{p\text{-}PIC}(F, opt{:}D))$ jest przeznaczony do magazynowania u sprzedawcy, do czasu etapu identyfikacji piratów. Sprzedawca nie ma dostępu do klucza deszyfrującego ani fingerprintu, ponieważ nie jest w stanie odszyfrować danych przeznaczonych dla klienta lub PIC.

Po odebraniu $\text{Sign}_{KGC}(\text{Encr}_{p\text{-}K}(D))$ oraz $\text{Sign}_{KGC}(\text{Encr}_{p\text{-}PIC}(F, opt{:}D))$ i pomyślnej weryfikacji podpisu, sprzedawca zapisuje do swojego repozytorium $\text{Sign}_{KGC}(\text{Encr}_{p\text{-}PIC}(F, opt{:}D))$ powiązane z Cert_K danego klienta. Następnie, sprzedawca przekazuje $\text{Sign}_{KGC}(\text{Encr}_{p\text{-}K}(D))$ klientowi (rys. 10.4. wiadomość 6) oraz dodaje go do grupy multicastowej, aby mógł on odebrać zaszyfrowany zamówiony obraz Y (rys. 10. wiadomość 7). Klient weryfikuje podpis na podstawie znanego mu certyfikat Cert_{KGC}, a następnie odszyfrowuje klucz D. W rezultacie, klient może zrealizować łączny fingerprinting i deszyfrację na obrazie Y, za pomocą klucza D.



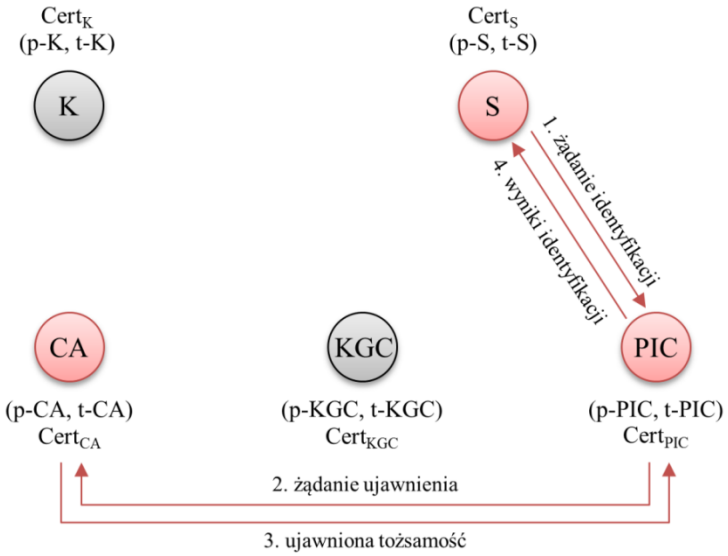
Rys. 10.3. Ogólny schemat etapu dystrybucji kluczy deszyfrujących, opracowanie własne.



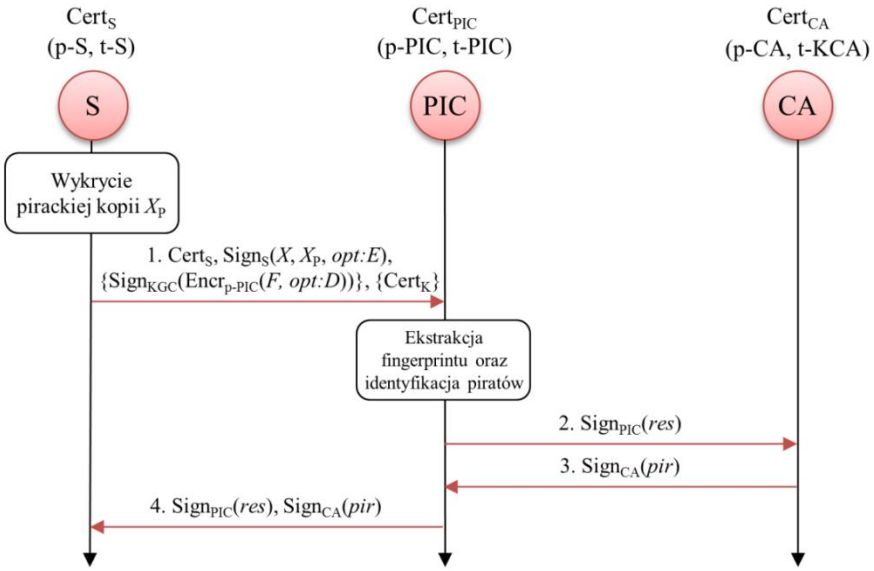
Rys. 10.4. Szczegółowy schemat dystrybucji kluczy deszyfrujących, opracowanie własne.

Kiedy sprzedawca wykryje, że istnieje piracka kopia X_P, to może zainicjować etap identyfikacji piratów. W tym celu, sprzedawca zgłasza do PIC żądanie identyfikacji (rys. 10.6. wiadomość 1), które zawiera dane: Cert_S, X, X_P, oraz opcjonalnie E, jeżeli wymaga tego wy-

korzystywana metoda JFD, a także zbiór F i opcjonalnie D zaszyfrowanych kluczem publicznym PIC z przypisanymi Cert_K każdego klienta, które sprzedawca otrzymał w etapie dystrybucji od KGC i przechowywał w swoim repozytorium. Dane X, X_P i E są podpisane przez sprzedawcę, natomiast zaszyfrowane F i D są podpisane przez KGC. PIC odbiera żądanie identyfikacji z wszystkimi przesłanymi danymi, po czym weryfikuje podpisy i odszyfrowuje zbiór fingerprintów F swoim kluczem tajnym. Następnie, posiadając wszystkie niezbędne dane, PIC przystępuje do ekstrakcji fingerprintu z pirackiej kopii oraz identyfikacji piratów, zgodnie z algorytmem wykorzystywanej metody JFD. W przypadku pomyślnej identyfikacji piratów, PIC podpisuje wyniki procesu identyfikacji i wysyła je do CA z żądaniem ujawnienia tożsamości piratów, jeżeli posługiwali się anonimowymi certyfikatami (rys. 10.6. wiadomość 2). CA weryfikuje podpis tego żądania i w przypadku pozytywnej weryfikacji wysyła do PIC podpisaną przez siebie prawdziwą tożsamość piratów (rys. 10.6. wiadomość 3).



Rys. 10.5. Ogólny schemat etapu identyfikacji piratów, opracowanie własne.



Rys. 10.6. Szczegółowy schemat etapu identyfikacji piratów, opracowanie własne.

W ostatnim kroku (rys. 10.6. wiadomość 4), PIC przekazuje do sprzedawcy wyniki identyfikacji podpisane przez PIC oraz tożsamość piratów podpisaną przez CA. W rezultacie, sprzedawca posiada dość informacji, aby móc domagać się kary dla piratów na drodze sądowej. Etap identyfikacji piratów został przedstawiony na rysunkach 10.5. i 10.6. Wiadomości 2 i 3 są pomijane, jeżeli klienci nie używali anonimowych certyfikatów.

Protokół jest sprawiedliwy, ponieważ zapewnia oczekiwane bezpieczeństwo zarówno z punktu widzenia sprzedawcy, jak i klienta. Z perspektywy sprzedawcy, protokół jest bezpieczny, gdyż klient nie ma dostępu do obrazu oryginalnego X oraz fingerprintu F, a zatem ochrona praw autorskich może być realizowana dzięki współpracującej metodzie JFD. Z perspektywy klienta, protokół jest bezpieczny, gdyż sprzedawca nie ma dostępu do fingerprintu F ani klucza deszyfrującego D i sprzedawca nie jest w stanie wygenerować oznakowanej kopii obrazu, a zatem problem praw klienta jest rozwiązany. Ponadto, jedna ze składowych fingerprintu F_{ord} jest zależna od parametrów zamówienia, co oznacza, że fingerprint nie może zostać przeniesiony do obrazu z innego zamówienia, a zatem problem niepowiązania jest rozwiązany.

Wzajemne uwierzytelnienie stron jest zrealizowane za pomocą certyfikatów i podpisów cyfrowych, dostępnych przy użyciu systemu PKI. Podpisy cyfrowe w proponowanym protokole służą uwierzytelnieniu źródła wiadomości. W przypadku, gdy podpis w dowolnej wiadomości nie przejdzie pomyślnie weryfikacji, należy przerwać działanie protokołu, gdyż oznacza to, że nadawca wiadomości podszywa się pod inną stronę. Ponadto, jeżeli podpis został zweryfikowany pomyślnie, ale wiadomość została podpisana przez inną stronę niż spodziewana, oznacza to, że nadawca wiadomości przedstawia dane, do których nie powinien mieć dostępu.

Protokół jest wygodny dla klienta, ponieważ kontaktuje się on wyłącznie ze sprzedawcą. Z perspektywy klienta, w celu uzyskania dostępu do usługi należy tylko zgłosić żądanie i wysłać swoją składową fingerprintu F_K i czekać na dostarczenie obrazu i klucza. Ponadto, prywatność klienta jest chroniona, dzięki wykorzystaniu anonimowych certyfikatów klientów. Z drugiej strony, współpraca z klientem nie jest wymagana podczas procesu identyfikacji piratów, tj. klient nie musi wysyłać do zaufanej trzeciej strony swojego klucza lub oznakowanej kopii, ani nie musi przedstawiać swojej prawdziwej tożsamości. Eliminuje to problemy, które mogłyby wyniknąć przy kliencie nieskłonnym do współpracy.

Ponieważ wykorzystywana jest metoda JFD, fingerprinty są osadzane po stronie klienta, a zatem KGC nie wymaga dużych zasobów obliczeniowych, ponieważ KGC jedynie generuje klucze deszyfrujące, a nie osadza fingerprintów w multimediach. Ponadto, obrazy nieoznakowane i zaszyfrowane fingerprinty są magazynowane przez sprzedawcę, zatem KGC i PIC nie wymagają dużych zasobów pamięciowych. Dodatkowo, możliwe jest fizyczne rozdzielenie KGC i PIC w celu optymalizacji obciążenia urządzeń, ponieważ ich funkcjonalność jest rozłączna. Jednakże, istotnym wymaganiem jest fakt, że KGC musi być dostępne, w sensie sieciowym, dla sprzedawcy w celu prowadzenia dystrybucji multimediów. Ponieważ dostępność KGC jest krytyczna dla świadczenia usługi, pożądanym byłoby, aby sprzedawca dysponował redundantnym dostępem do sieci.

10.3. Podsumowanie

W rozdziale przedstawiono problemy związane z brakiem ochrony klienta występującym podczas stosowania metod łącznego fingerprintingu i deszyfracji. Głównymi zagrożeniami rozpatrywanymi w tym rozdziale był problem praw klienta (ang. *custormer's right problem*) oraz problem niepowiązania (ang. *unbinding problem*), które powodują, że nieuczciwy sprzedawca jest w stanie działać na szkodę uczciwego klienta. W przypadku problemu praw klienta, strona dystrybucyjna jest w stanie wygenerować oznakowaną kopię multimediów, udostępnić ją w sieci i wrobić uczciwego klienta w piractwo. W przypadku problemu niepowiązania, strona dystrybucyjna jest w stanie przenieść fingerprint z faktycznej pirackiej kopii do innej treści multimedialnej, co zaowocowałoby większym odszkodowaniem w przypadku skazania pirata.

Zaproponowano sprawiedliwy protokół zarządzania dystrybucją dla metod JFD, którego przeznaczeniem jest uniknięcie powyższych zagrożeń. Protokół ten wykorzystuje zaufaną trzecią stronę, podzieloną funkcjonalnie na centrum generowania kluczy i centrum identyfikacji piratów, oraz działa w trzech etapach: rejestracji anonimowych klientów, dystrybucji kluczy deszyfrujących oraz identyfikacji piratów. Proponowane rozwiązanie może współpracować z nowymi metodami JFD przedstawionymi w niniejszej rozprawie oraz innymi metodami, które są zgodne z koncepcją JFD. Protokół ten jest sprawiedliwy, ponieważ zapewnia oczekiwane bezpieczeństwo zarówno z perspektywy sprzedawcy, jak i klienta. Wzajemne uwierzytelnienie stron jest zrealizowane za pomocą certyfikatów i podpisów cyfrowych. Protokół wymaga od klienta kontaktu jedynie ze sprzedawcą i jednocześnie nie wymaga współpracy z klientem podczas identyfikacji piratów.

11. PODSUMOWANIE

Problemem badawczym, którego dotyczyła niniejsza rozprawa jest przeciwdziałanie zjawisku nielegalnego rozpowszechniania multimediów, bez zgody ich autorów, czyli przeciwdziałanie piractwu. Konieczne jest opracowanie rozwiązań, które pozwolą na bezpieczne i ekonomiczne przesyłanie treści multimedialnych przez sieć. Rozwiązania te powinny gwarantować, że multimedia będą używane zgodnie z przeznaczeniem i tylko przez uprawnionych użytkowników, którzy uiszczą odpowiednie opłaty. Problem ten został już sformułowany i znane są rozwiązania wykorzystujące techniki cyfrowego odcisku palca, polegające na osadzaniu ukrytych danych w multimediach, których obecność ma służyć identyfikacji użytkowników biorących udział w nielegalnej redystrybucji. Jednakże, dotychczasowe metody fingerprintingu okazały się nieekonomiczne ze względu na wymagane pasmo oraz moc obliczeniową. W ostatnich latach nastąpił zwrot w koncepcji fingerprintingu w postaci metod łącznego fingerprintingu i deszyfracji JFD, które charakteryzują się minimalnymi zapotrzebowaniami na zasoby sieciowe i obliczeniowe. Jednakże, istniejące rozwiązania zapewniają niewystarczającą odporność na ataki piratów mające na celu usunięcie fingerprintu z obrazu. Ponadto, dotychczas zaproponowane rozwiązania zostały zaprojektowane dla obrazów w odcieniach szarości, a ich przystosowanie do obrazów kolorowych spowodowałoby znaczny wzrost wymagań na zasoby obliczeniowe, gdyż każdy kolor byłby przetwarzany niezależnie. Dotychczas, w dostępnej literaturze nie są znane metody JFD, które przetwarzałyby wszystkie kolory jednocześnie we wspólnej trójwymiarowej przestrzeni, która może być zamodelowana za pomocą kwaternionów. Wobec tego, konieczne jest opracowanie nowych metod JFD dla obrazów kolorowych, które będą ekonomiczne pod względem wykorzystania zasobów i będą zapewniać wysoką odporność na próby usunięcia fingerprintów.

Celem pracy było zaprojektowanie skutecznej metody łącznego fingerprintingu i deszyfracji dla obrazów kolorowych wykorzystującej rachunek kwaternionowy. Metoda miała obejmować trzy algorytmy: algorytm dla strony nadawczej realizujący konstrukcję kluczy i selektywne szyfrowanie obrazu; algorytm dla strony odbiorczej realizujący deszyfrowanie oraz osadzanie cyfrowych odcisków palca; oraz algorytm detekcji dla strony nadawczej lub zaufanej trzeciej strony przeznaczony do identyfikacji użytkowników biorących udział w nielegalnej redystrybucji obrazu. Cyfrowe odciski palca miały być osadzane w obrazie w sposób niezauważalny dla ludzkiego oka. Projektowana metoda miała być odporna na próby usunięcia osadzanych fingerprintów. Głównym kryterium oceny skuteczności projektowanej metody miała być odporność na ataki zmowy, czyli ataki realizowane przez zorganizowane grupy nieuczciwych użytkowników. Cel pracy został zrealizowany. Przedstawiono trzy nowe metody JFD spełniające powyższe założenia, z czego dwie wykorzystują rachunek kwaternionowy. Algorytmy tych metod zamieszczono w rozdziałach 5., 7. i 8., niezauważalność osadzonych fingerprintów wykazano w podrozdziale 9.2., a odporność na próby usunięcia fingerprintów wykazano w podrozdziałach od 9.3. do 9.6. Teza pracy została sformułowana następująco: Wykorzystanie kwaternionowej reprezentacji obrazów kolorowych i właściwości rachunku kwaternionowego umożliwia zrealizowanie systemu łącznego fingerprintingu i deszyfracji odpornego na ataki zmowy. Wyniki badań przedstawione w podrozdziałach 9.5. i 9.6., dla metody opartej na rotacji kwaternionów z wiązaniem kwaternionów zaszyfrowanych oraz dla metody opartej na serii rotacji kwaternionów, wykazały prawdziwość tezy.

W części wstępnej rozprawy, opisano podstawowe zagadnienia dotyczące ochrony multimediów poprzez fingerprinting. Najpierw, przedstawiono klasyczny model fingerprintingu, kryteria konstrukcji zbioru fingerprintów, wymagania stawiane metodom fingerprintingu, oraz podział tych metod ze względu na umiejscowienie operacji osadzania fingerprintu w danych. Następnie, przedstawiono zagrożenia dla osadzonych fingerprintów z podziałem na ataki realizowane przez pojedynczego pirata oraz ataki realizowane przez grupę piratów, czyli ataki zmowy. Omówiono również korzyści wynikające z osadzania fingerprintów w dziedzinie częstotliwości zamiast w dziedzinie przestrzennej. Kolejnym etapem było przedstawienie koncepcji łącznego fingerprintingu i deszyfracji. Przedstawiono ogólne zasady działania metod JFD, omówiono bezpieczeństwo szyfrowania w JFD oraz przedstawiono przegląd literatury metod JFD związanych z rozwojem nowych metod JFD przedstawionych w tej rozprawie.

W części głównej rozprawy, przedstawiono oryginalny dorobek autora, na który składają się: nowa metoda JFD oparta na iloczynie macierzy, przedstawiona w rozdziale 5., nowa metoda JFD oparta na rotacji kwaternionów w trybie wiązania kwaternionów zaszyfrowanych, przedstawiona w rozdziale 7., nowa metoda JFD oparta na serii rotacji kwaternionów, przedstawiona w rozdziale 8., wyniki badań zaproponowanych metod, zawarte w rozdziale 9., oraz sprawiedliwy protokół zarządzania dystrybucją dla metod JFD, opisany w rozdziale 10. Dla każdej nowej metody przedstawiono opis konstrukcji kluczy, algorytm szyfrowania, algorytm deszyfrowania z jednoczesnym osadzeniem fingerprintu, a także algorytm identyfikacji piratów. Kody źródłowe każdej z metod, napisane w środowisku Matlab, zostały zamieszczone w załączniku C. Stworzone programy symulują system dystrybucji treści multimedialnych, w którym dane są szyfrowane wspólnym kluczem i przesyłane do wszystkich odbiorców za pomocą jednego połączenia multicast, natomiast klucze deszyfrujące są dostarczane indywidualną drogą do każdego odbiorcy poprzez połączenia unicast. Dla każdej nowej metody przeprowadzono badania, które miały na celu sprawdzenie niezauważalności osadzonych fingerprintów oraz ich odporności na szum, kompresję, ataki zmowy i ataki zmowy połączone z kompresją. Zaimplementowane metody realizują znakowanie obrazów nieruchomych. Jednakże, opracowany algorytm postępowania może być w ogólności wykorzystany do znakowania sekwencji video, w których wybrane ramki są przetwarzane analogicznie do obrazów nieruchomych. Dlatego też, wyniki badań są uniwersalne i słuszne również dla znakowania strumieni wideo. Następnie, przedstawiono porównanie nowych metod JFD z istniejącymi rozwiązaniami z literatury oraz wielokryterialnie oceniono każde rozwiązanie. Na końcu rozprawy, zaproponowano sprawiedliwy protokół zarządzania dystrybucją dla metod JFD rozwiązujący problem praw klienta oraz problem niepowiązania.

W rozprawie przedstawiono ocenę nowych metod JFD na podstawie wyników badań właściwości tych metod oraz porównania z innymi rozwiązaniami. Metody zostały ocenione pod względem niezauważalności osadzanych fingerprintów, odporności na szum, odporności na kompresję, odporności na ataki zmowy, odporności na ataki zmowy z kompresją, wydajności konstrukcji, wydajności dystrybucji oraz wydajności detekcji. Ponadto, zauważono następujące ograniczenia proponowanych rozwiązań: konieczność implementacji operacji algebry kwaternionów lub wykorzystanie odpowiedniej biblioteki oraz niska wydajności detekcji w porównaniu do metod osadzających fingerprinty w sposób addytywny. Pomimo istniejących ograniczeń, proponowane nowe metody JFD dla obrazów kolorowych mogą być stosowane w usługach wideo na żądanie i strumieniowania wideo dla zapewnienia ochrony praw

autorskich poprzez skuteczną identyfikację piratów przy jednoczesnej ekonomicznej dystrybucji multimediów. Niniejsza rozprawa zawiera treści stanowiące znaczący wkład do rozwoju dziedziny cyfrowego znakowania danych, w szczególności cyfrowych odcisków palców. Metoda JFD oparta na rotacji kwaternionów z wiązaniem kwaternionów zaszyfrowanych oraz metoda JFD oparta na serii rotacji kwaternionów przedstawiają innowacyjne wykorzystanie kwaternionów do celów łącznego fingerprintingu i deszyfracji. Osadzanie fingerprintów w trójwymiarowej przestrzeni kolorów jest oryginalnym podejściem w tej dziedzinie nauki. Ponadto, zaproponowany sprawiedliwy protokół dla ochrony klienta może być wykorzystany nie tylko wraz z nowymi metodami z niniejszej rozprawy, ale również z dowolną metodą, zgodną z koncepcją łącznego fingerprintingu i deszyfracji. Do powstania nowych metod JFD i osiągnięcia powyższych rezultatów doprowadziły prace nad publikacjami [29, 30, 37, 86, 88, 89, 90, 91, 92, 93, 96, 97, 98], które stanowią oryginalny dorobek autora.

Na zakończenie należy rozpatrzyć, jakie inne podejścia lub założenia mógł przyjąć autor rozprawy oraz jakie nowe ścieżki badań otwiera rozprawa. Biorąc pod uwagę stan wiedzy autora po zakończeniu badań, założenia projektowe dla nowych metod JFD mogłyby się różnić w trzech obszarach: wyznaczanie współczynnika siły osadzania fingerprintów, stosowanie ciągów odpornych na zmowę, alternatywna ekstrakcja fingerprintu. Po pierwsze, w celu osiągnięcia niezauważalności fingerprintów, współczynnik siły osadzania był wybierany eksperymentalnie, z założeniem uzyskania stosunku PSNR o wartości przynajmniej 40 dB. Zamiast eksperymentalnego wyboru, współczynnik siły osadzania fingerprintów mógłby być wyznaczany za pomocą algorytmu uwzględniającego model ludzkiej percepcji HVS (ang. *Human Visual System*) i lokalne właściwości obrazu. Potencjalnie, mogłoby to doprowadzić do jeszcze lepszej niezauważalności osadzonych fingerprintów. Po drugie, podczas badań w roli fingerprintów stosowano różne ciągi pseudolosowe, tj. ciągi o rozkładzie normalnym, ciągi o rozkładzie normalnym, *m*-ciągi, ciągi Golda, ciągi Kasami i ciągi Hadamarda. Alternatywnie, możliwe było przyjęcie założenia wykorzystania fingerprintów odpornych na ataki zmowy, tzn. kodów Tardos lub kodów Boneh-Shaw. W rezultacie, nowe metody JFD znacznie by się różniły, ponieważ musiałyby umożliwiać zastosowanie takiej kodowej postaci fingerprintów, tzn. konieczna byłaby ekstrakcja fingeprintu w dokładnie takiej samej postaci, w jakiej był dodany do klucza. Po trzecie, w proponowanych metodach w wyniku ekstrakcji fingerprintu, uzyskiwana jest inna postać fingerprintu niż ciąg dodany do klucza deszyfrującego. Z tego względu, w celu identyfikacji piratów, konieczne jest obliczenie właściwej postaci osadzonego fingerprintu dla każdego użytkownika, co pogarsza wydajność detekcji. Możliwe jest przyjęcie innego podejścia i zaprojektowanie takiego algorytmu ekstrakcji fingerprintu, aby wydobyty fingerprint był w takiej samej postaci, jak ten dodawany do klucza.

Poza badaniami zasugerowanymi przez powyższą krytyczną refleksję, istnieją jeszcze nowe kierunki badań uwzględniające inne aspekty bezpieczeństwa, które nie zostały poruszone w rozprawie. Na przykład, nowe metody JFD nie zostały zbadane pod kątem odporności na atak zmowy na klucze deszyfrujące. Analogicznie do ataków zmowy na oznakowanych obrazach, piraci mogą podjąć próbę wygenerowania pirackiego klucza deszyfrującego na podstawie zbioru swoich legalnie pozyskanych kluczy, np. uśredniając swoje klucze deszyfrujące. Piracka kopia obrazu może powstać w wyniku deszyfrowania obrazu za pomocą takiego pirackiego klucza. Wobec tego, istnieje perspektywa badania odporności nowych metod JFD na ataki zmowy na klucze deszyfrujące. Dodatkowo, abstrahując od właściwości nowych me-

tod JFD, w przypadku każdej metody JFD konieczne jest bezpieczne dostarczenie kluczy deszyfrujących do wszystkich odbiorców usługi. Ten aspekt systemu nie był analizowany w rozprawie i może być tematem dalszych badań. Pomimo, że w metodach łącznego fingerprintingu i deszyfracji, głównym celem jest osiągnięcie skutecznej identyfikacji piratów (ang. *traitor tracing*), a nie bezpieczeństwo szyfrowania, to kolejnym obszarem nowych badań może być analiza szyfrogramów otrzymanych w nowych metodach JFD, np. analiza histogramów lub pomiar PSNR szyfrogramów. Ponadto, innowacyjne wykorzystanie kwaternionów oraz operacji w trójwymiarowej przestrzeni kolorów, zaprezentowane w tej rozprawie, pokazuje zupełnie nowe podejście do projektowania metod fingerprintingu i otwiera nowe perspektywy przyszłych prac naukowo-badawczych.

BIBLIOGRAFIA

[1] „YouTube Statistics," 18 10 2014. [Online]. Available: https://www.youtube.com/yt/press/statistics.html.

[2] „Business Insider," 18 10 2014. [Online]. Available: http://www.businessinsider.com/amazon-buys-twitch-2014-8.

[3] K. Liu, W. Trappe, Z. Wang, M. Wu i H. Zhao, Multimedia fingerprinting forensics for traitor tracing, EURASIP Book Series on Signal Processing and Communications red., tom 4, Hindawi Publishing Corporation, 2005.

[4] K. Liu, W. Trappe, Z. Wang i M. Wu, „Collusion-resistant fingerprinting for multimedia," *IEEE Signal Processing Magazine*, tom 21, s. 15-27, 2004.

[5] K. Liu, Z. Wang, M. Wu i H. Zhao, „Forensic analysis of nonlinear collusion attacks for multimedia fingerprinting," *IEEE Transactions on Image Processing*, tom 14, nr 5, s. 646-661, 2005.

[6] S. He i M. Wu, „Joint coding and embedding techniques for multimedia fingerprinting," *IEEE Transactions on Information Forensics and Security*, tom 1, nr 2, s. 231-247, 2006.

[7] M. Wu, W. Trappe, Z. Wang i K. Liu, „Collusion resistant multimedia fingerprinting: a unified framework," *Proc. SPIE 5306, Security, Steganography, and Watermarking of Multimedia Contents VI*, 2004.

[8] X. Li, B. Guo i X. Wu, „On collusion attack for digital fingerprinting," *Journal of Information Hiding and Multimedia Signal Processing*, tom 2, nr 4, s. 366-376, 2011.

[9] D. Kundur i K. Karthik, „Video fingerprinting and encryption principles for digital rights management," *Proceedings of the IEEE*, tom 92, nr 6, s. 918-932, 2004.

[10] R. Anderson i C. Manifavas, „Chameleon - a new kind of stream cipher," w *Lecture Notes in Computer Science, Fast Software Encryption*, E. Biham, Red., Springer-Verlag, 1997, s. 107-113.

[11] A. Adelsbach, U. Huber i A. Sadeghi, „Fingercasting - joint fingerprinting and decryption of broadcast messages," *11th Australasian Conference on Information Security Privacy*, tom 4058, s. 136-147, 2006.

[12] A. Piva, T. Bianchi i A. De Rosa, „Secure client-sde ST-DM watermark embedding," *IEEE Transactions on Information Forensics and Security*, tom 5, nr 1, s. 13-26, 2010.

[13] R. Rykaczewski, „Hillcast – metoda łącznego kryptograficznego zabezpieczenia danych dla multicastowej dystrybucji informacji," *Zeszyty Naukowe Wydziału ETI Politechniki Gdańskiej,* tom 19, nr 8, s. 231-236, 2010.

[14] F. Zhang, „Quaternion and matrices of quaternions," w *Linear Algebra and its Applications*, Elsevier Science Inc., 1997, s. 21-57.

[15] M. Baker, „Euclidean Space," 15 10 2014. [Online]. Available: http://www.euclideanspace.com/maths/algebra/realNormedAlgebra/quaternions/index.htm.

[16] H. Zhao i K. Liu, „Bandwidth efficient fingerprint multicast for video streaming," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04),* tom 5, s. 849-852, 2004.

[17] J. Blinn, „What's the deal with the DCT?," *IEEE Computer Graphics and Applications*, tom 13, nr 4, s. 78-83, 1993.

[18] K. Cabeen i P. Gent, „Image compression and the discrete cosine transform," *College of the Redwoods*, 1998.

[19] L. Tang, „Methods for encrypting and decrypting MPEG video data efficiently," *4th ACM International Conference on Multimedia (MULTIMEDIA '96)*, s. 219-229, 1996.

[20] I. Cox, J. Kilian, F. Leighton i T. Shamoon, „Secure spread spectrum watermarking for multimedia," *IEEE Transactions on Image Processing*, tom 6, nr 12, s. 1673-1687, 1997.

[21] J. Kilian, F. Leighton, L. Matheson, T. Shamoon, R. Tarjan i F. Zane, „Resistance of digital watermarks to collusive attacks," *IEEE International Symposium on Information Theory*, 1998.

[22] A. Varna, S. He, A. Swaminathan i M. Wu, „Fingerprinting compressed multimedia signals," *IEEE Transactions on Information Forensics and Security*, tom 4, nr 3, s. 330-345, 2009.

[23] D. Kedia, „Comparative analysis of peak correlation characteristics of non-orthogonal spreading codes for wireless systems," *International Journal of Distributed and Parallel Systems*, tom 3, nr 3, s. 63-74, 2012.

[24] A. Kortun i A. Hocanin, „Spreading codes in CDMA detection," *Course of EE-574 Detection & Estimation Theory*, 2003.

[25] K. Kettunen, „Code selection for CDMA systems," *Department of Information Studies, University of Tampere,* 1997.

[26] X. Zeng, Q. Liu i L. Hu, „A new family of codes and generalized kasami sequences," *IEEE International Symposium on Information Theory,* s. 907-911, 2006.

[27] Y. Tsai i X. Li., „Kasami code-shift-keying modulation for ultra wideband communication systems," *IEEE 2006 International Conference on Ultra-Wideband,* s. 37-42, 2006.

[28] A. Chandra i S. Chattopadhyay, „Small set orthogonal Kasami codes for CDMA system," *4th International Conference on Computer and Devices for Communication (CODEC 2009),* s. 1-4, 2009.

[29] B. Czaplewski i R. Rykaczewski, „Łączony fingerprinting i kryptograficzne zabezpieczenie danych z wykorzystaniem szyfru Hilla," *Zeszyty Naukowe Wydziału Elektroniki, Telekomunikacji i Informatyki Politechniki Gdańskiej,* tom 1, nr 9, s. 237-242, 2011.

[30] B. Czaplewski i R. Rykaczewski, „Improvement of fingerprinting method based on Hill cipher by using frequency domain," *ICT Young 2012 Conference,* s. 231-236, 2012.

[31] G. Tardos, „Optimal probabilistic fingerprint codes," *Journal of the ACM,* tom 55, nr 2, s. 116-125, 2008.

[32] D. Boneh i J. Shaw, „Collusion-secure fingerprinting for digital data," *IEEE Transactions on Information Theory,* tom 44, s. 1897-1905, 1998.

[33] B. Škorić i J. Oosterwijk, „Binary and q -ary Tardos codes, revisited," *Designs, Codes and Cryptography,* 2013.

[34] S. Ramesh i A. Shanmugam, „Compressed-domain watermarking algorithms: a review," *International Journal of Computer Science and Technology,* tom 2, nr 1, s. 101-105, 2011.

[35] H. Shan i W. Min, „Collusion-resistant video fingerprinting for large user group," *IEEE Transactions on Information Forensics and Security,* tom 2, nr 4, s. 697-709, 2007.

[36] S. Sun, C. Lu i P. Chang, „Joint multimedia fingerprinting and encryption: security issues and some solutions," *IEEE International Conference on Multimedia and Expo,* s. 1531-1534, 2007.

[37] B. Czaplewski, M. Dzwonkowski i R. Rykaczewski, „Digital fingerprinting for color images based on the quaternion encryption scheme," *Pattern Recognition Letters,* tom 46, s. 11-19, 2014.

[38] J. Domingo-Ferrer, F. Sebe i A. Martinez-Balleste, „On multicast fingerprinting and collusion security," *1st International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution (AXMEDIS 2005),* 2005.

[39] X. Chen i S. Lian, „On the collusion-resistance of the fingerprinting algorithm based on additive watermarking," *Journal of Multimedia,* tom 6, nr 4, s. 369-375.

[40] M. Kuribayashi, „Hierarchical spread spectrum fingerprinting scheme based on the CDMA technique," *EURASIP Journal on Information Security,* tom 2011:502782, 2011.

[41] H. Schaathun, „Attacks on Kuribayashi's fingerprinting scheme," *IEEE Transactions on Information Forensics and Security,* tom 9, nr 4, s. 607-609, 2014.

[42] S. Sun, C. Lu i P. Chang, „AACS-compatible multimedia joint encryption and fingerprinting: security issues and some solutions," *Signal Processing: Image Communication,* tom 23, nr 3, s. 179-193, 2008.

[43] D. Bouslimi, G. Coatrieux, M. Cozic i C. Roux, „A joint encryption/watermarking system for verifying the reliability of medical images," *IEEE Transactions on Information Technology in Biomedicine,* tom 16, nr 5, s. 891-899, 2012.

[44] F. Battisti, M. Cancellaro, G. Boato, M. Carli i A. Neri, „Joint watermarking and encryption of color images in the Fibonacci-Haar domain," *EURASIP Journal on Advances in Signal Processing,* s. 1-13, 2009.

[45] M. Cancellaro, F. Battisti, M. Carli, G. Boato, F. De Natale i A. Neri, „A joint digital watermarking and encryption method," *SPIE 6819, Security, Forensics, Steganography, and Watermarking of Multimedia, Contents X,* s. 68191C, 2008.

[46] S. Paul, Multicast on the Internet and its Application, Kluwer Academic, 1998.

[47] T. Bartczak, M. Piechowiak, T. Szewczyk i P. Zwierzykowski, „Multicast IP - Technologia przyszłości," *Networld,* 2007.

[48] M. Amma i P. Judge, „WHIM: Watermarking multicast video with a hierarchy of intermediaries," *International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '00),* 2000.

[49] J. Bloom, „Security and rights management in digital cinema," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '03),* tom 4, s. 712-715, 2003.

[50] Y. Lin, C. Huang i G. Lee, „Rotation, scaling, and translation resilient watermarking for images," *IET Image Processing,* tom 5, nr 4, s. 328-340, 2011.

[51] G. Doerr i J. Dugelay, „New intra-video collusion attack using mosaicing," *International Conference on Multimedia and Expo ICME '03*, tom 2, s. 505-508, 2003.

[52] A. Houmansadr i S. Ghaemmaghami, „A collusion-resistant video watermarking scheme," *Information Hiding, Lecture Notes in Computer Science,* tom 4437, s. 343-354, 2007.

[53] A. Boroujerdizadeh, A. Ghobadi, A. Yaribakht i M. Shahidan Bin Abdoullah, „A novel method to reduce collusion attack possibility on watermarking," *15th International Conference on Advanced Communication Technology (ICACT),* s. 1066 - 1071, 2013.

[54] H. Feng, H. Ling, F. Zou, W. Yan, M. Sarem i Z. Lu, „A collusion attack optimization framework toward spread-spectrum fingerprinting," *Applied Soft Computing,* tom 13, nr 8, s. 3482-3493, 2013.

[55] J. Fridrich, Steganography in digital media: principles, algorithms, and applications, Nowy Jork: Cambridge University Press, 2009.

[56] Y. Wang i A. Pearmain, „Blind image data hiding based on self reference," *Pattern Recognition Letters,* tom 25, nr 15, s. 1681-1689, 2004.

[57] C. Chang, Y. Hu i T. Lu, „A watermarking-based image ownership and tampering authentication scheme," *Pattern Recognition Letters,* tom 27, nr 5, s. 439-446, 2006.

[58] N. Jayant, J. Johnston i R. Safranek, „Signal compression based on models of human perception," *Proceedings of the IEEE,* tom 81, nr 10, 1993.

[59] S. Suthaharan, S. Kim, H. Lee i S. Sathananthan, „Perceptually tuned robust watermarking scheme for digital images," *Pattern Recognition Letters,* tom 21, nr 2, s. 145-149, 2000.

[60] V. Mehan, R. Dhir i Y. Brar, „Joint watermarking and fingerprinting approach for colored digital image in double DCT domain," *IEEE International Conference on Signal Processing, Computing and Control (IEEE ISPCC),* s. 1-6, 2013.

[61] X. Li, B. Guo, F. Meng i L. Li, „A novel fingerprinting algorithm with blind detection in DCT domain for images," *AEU - International Journal of Electronics and Communications,* tom 65, nr 11, s. 942-948, 2011.

[62] S. Maity, M. Kundu i T. Das, „Robust SS watermarking with improved capacity," *Pattern Recognition Letters,* tom 28, nr 3, s. 350-356, 2007.

[63] Y. Kim, O. Kwon i R. Park, „Wavelet based watermarking method for digital images using the human visual system," *Electronic Letters,* tom 35, nr 6, s. 466-468, 1999.

[64] S. Vural, „Video watermarking for digital cinema contents," *13th European Signal Processing Conference,* s. 303-304, 2005.

[65] C. Podilchuk, „Image-adaptive watermarking using visual models," *IEEE Journal on Selected Areas in Communications,* tom 16, nr 4, s. 525-539, 1998.

[66] C. Lee, H. Oh, Y. Baek i H. Lee, „Adaptive digital image watermarking using variable size of blocks in frequency domain," *IEEE Region 10 Conference (TENCON 99),* tom 1, s. 702-705, 1999.

[67] S. Hou, T. Uehara, T. Satoh i Y. Morimura, „Integrating fingerprint with cryptosystem for internet-based live pay-TV system," *ChinaCom2008, Third International Conference on Communications and Networking in China,* s. 1281 - 1286, 2008.

[68] T. Thanh i M. Iwakiri, „Nonspecific DCT-block fingerprinting based on incomplete cryptography for DRM system," *International Conference on Advanced Technologies for Communications (ATC 2012),* s. 300-303, 2012.

[69] H. Cheng i X. Li, „Partial encryption of compressed images and videos," *IEEE Transactions on Signal Processing,* tom 48, s. 2439-2451, 2000.

[70] G. Bhatnagar i Q. Wu, „Selective image encryption based on pixels of interest and singular value decomposition," *Digital Signal Processing,* tom 22, s. 648-663, 2012.

[71] X. Chen i S. Lian, „On the DCT based joint fingerprint embedding and decryption scheme," *Journal of Multimedia,* tom 5, nr 5, s. 473-480, 2010.

[72] S. Katzenbeisser, B. Škoric, M. Celik i A. Sadeghi, „Combining Tardos fingerprinting codes and fingercasting," *Information Hiding, Lecture Notes in Computer Science,* tom 4567, s. 294-310, 2007.

[73] A. Lemma, S. Katzenbeisser, M. Celik i M. Veen, „Secure watermark embedding through partial encryption," *International Workshop on Digital Watermarking (IWDW 2006),* tom 4283, s. 433-445, 2006.

[74] S. Lian, X. Chen, Y. Dong i H. Wang, „On the secure multimedia distribution scheme based on partial

encryption," *IEEE International Conference on Communications (ICC),* s. 1-5, 2010.

[75] S. Lian, Z. Liu i H. Wang, „On the joint audio fingerprinting and decryption scheme," *IEEE International Conference on Multimedia and Expo,* s. 261-264, 2008.

[76] M. Celik, A. Lemma, S. Katzenbeisser i M. van der Veen, „Lookup-table-based secure client-side embedding for spread-spectrum watermarks," *IEEE Transactions of Information Forensics and Security,* tom 3, nr 3, s. 475-487, 2008.

[77] C. Lin, P. Prangjarote, L. Kang, W. Huang i T. Chen, „Joint fingerprinting and decryption with noise-resistant for vector quantization images," *Signal Processing,* tom 92, nr 9, s. 2159-2171, 2012.

[78] P. Prangjarote, C. Lin, L. Kang i C. Yeh, „Joint fingerprinting and decryption for VQ images through bipartite matching," *Sixth International Conference on Genetic and Evolutionary Computing (ICGEC),* s. 27-30, 2012.

[79] M. Li, D. Xial, Y. Zhang i H. Liu, „Attack and improvement of the joint fingerprinting and decryption method for vector quantization images," *Signal Processing,* tom 99, s. 17-28, 2014.

[80] C. Lin, P. Prangjarote, C. Yeh i H. Ng, „Reversible joint fingerprinting and decryption based on side match vector quantization," *Signal Processing,* tom 98, s. 52-61, 2014.

[81] C. Lin, W. Huang i T. Chen, „Noise-resistant joint fingerprinting and decryption based on vector quantization," *International Conference on Broadband, Wireless Computing, Communication and Applications,* s. 463 - 468, 2010.

[82] R. Parnes i R. Parviainen, „Large scale distributed watermarking of multicast media through encryption," *IFIP International Conference on Communications and Multimedia Security Issues of the New Century,* s. 17, 2001.

[83] Y. Xu, L. Xioung, Z. Xu i S. Pan, „A content security protection scheme in JPEG compressed domain," *Journal of Visual Communication and Image Representation,* tom 25, nr 5, s. 805-813, 2014.

[84] S. Lian, Z. Liu, Z. Ren i H. Wang, „Joint fingerprint embedding and decryption for video distribution," *IEEE International Conference on Multimedia and Expo,* s. 1523-1526, 2007.

[85] Y. Chen, H. Huang i S. Wang, „Video scrambling and fingerprinting for digital right protection," *International symposium on computer, consumer and control (IS3C 2012),* s. 471-474, 2012.

[86] B. Czaplewski, „Implementacja i badanie metody fingercastingu wykorzystującej uogólniony szyfr Hilla," *Praca dyplomowa magisterska, Wydział ETI Politechniki Gdańskiej,* 2011.

[87] L. Wu, J. Zhang, W. Deng i D. He, „Arnold Transformation Algorithm and Anti-Arnold Transformation Algorithm," *1st International Conference on Information Science and Engineering (ICISE2009),* s. 1164-1167, 2009.

[88] B. Czaplewski i K. Czaplewski, „Protection of visual data transmission for vessel traffic systems using joint fingerprinting and decryption method based on modified Hill cipher," *Annual of Navigation,* tom 19, nr 2, s. 5-17, 2012.

[89] B. Czaplewski i K. Czaplewski, „Secure transmission of visual image in the VTS system using fingerprinting," *Scientific Journals of Szczecin Maritime University,* tom 30, nr 102, s. 30-36, 2012.

[90] B. Czaplewski i K. Czaplewski, „Protection of visual data transmission using joint fingerprinting and decryption method based on modified Hill cipher for vessel traffic systems," *The European Navigation Conference,* 2012.

[91] B. Czaplewski i R. Rykaczewski, „Ochrona usług masowej dystrybucji multimediów z wykorzystaniem metod fingercastingu," w *Nowoczesne systemy łączności i transmisji danych na rzecz bezpieczeństwa. Szanse i zagrożenia.*, Wolters Kluwer Polska SA, 2013, s. 94-124.

[92] B. Czaplewski i R. Rykaczewski, „Immunizing the Hillcast method against the known-plaintext attack using periodic key exchange," *Przegląd Telekomunikacyjny + Wiadomości Telekomunikacyjne,* nr 8-9, s. 792-798, 2013.

[93] B. Czaplewski i R. Rykaczewski, „Matrix-based robust joint fingerprinting and decryption method for multicast distribution of multimedia," *Signal Processing,* tom 111, s. 150-164, 2015.

[94] S. Saeednia, „How to make the Hill cipher secure," *Cryptologia,* tom 24, nr 4, s. 353-360, 2000.

[95] M. Toorani i A. Falahati, „A secure variant of the Hill cipher," *IEEE Symposium on Computers and Communications (ISCC 2009),* s. 313-316, 2009.

[96] B. Czaplewski, M. Dzwonkowski i R. Rykaczewski, „Digital fingerprinting based on quaternion encryption scheme for gray-tone images," *Journal of Telecommunications and Information Technology,* tom 2, s. 3-11, 2014.

[97] B. Czaplewski, M. Dzwonkowski i R. Rykaczewski, „Digital fingerprinting based on quaternion encryption for image transmission," *Przegląd Telekomunikacyjny + Wiadomości Telekomunikacyjne,* nr 8-9, s. 792-798, 2013.

[98] B. Czaplewski i R. Rykaczewski, „Receiver-side fingerprinting method based on quaternion sandwich product," *Przegląd Telekomunikacyjny + Wiadomości Telekomunikacyjne,* nr 8-9, s. 1170-1176, 2014.

[99] S. Sangwine i N. Le Bihan, „Quaternion and octonion toolbox for Matlab," 27 7 2005. [Online]. Available: http://qtfm.sourceforge.net/.

[100] S. Emmanuel i M. Kankanhalli, „A digital rights management scheme for broadcast video," *Multimedia Systems,* tom 8, s. 444-458, 2003.

[101] C. Lei, P. Yu, P. Tsai i M. Chan, „An efficient and anonymous buyer-seller watermarking protocol," *IEEE Transactions on Image Processing,* tom 13, nr 12, s. 1618-1626, 2004.

[102] J. Zhang, W. Kou i K. Fan, „Secure buyer-seller watermarking protocol," *IEE Proc. Information Security,* tom 153, nr 1, s. 15-18, 2006.

[103] F. Frattolillo, „Watermarking protocol for web context," *IEEE Transactions on Information Forensics and Security,* tom 2, nr 3, s. 350-363, 2007.

WYKAZ RYSUNKÓW I TABEL

Rys. 2.1. Klasyczny model fingerprintingu, opracowanie własne.

Rys. 2.2. Podział metod fingerprintingu ze względu na umiejscowienie procesu osadzania fingerprintów do danych multimedialnych, opracowanie własne.

Rys. 3.1. Podział zagrożeń dla osadzonych fingerprinty, opracowanie własne.

Rys. 3.2. a) Przykładowa charakterystyka ROC, b) charakterystyka ROC nieskutecznego detektora, c) charakterystyka ROC skutecznego detektora, d) charakterystyki ROC detektorów trudnych do porównania, opracowanie własne na podstawie [55].

Rys. 4.1. Schemat osadzania fingerprintu po stronie odbiorcy, opracowanie własne.

Rys. 4.2. Ogólna zasada działania łącznego fingerprintingu i deszyfracji, opracowanie własne.

Rys. 4.3. Koncepcja częściowego szyfrowania wideo, opracowanie własne na podstawie [9].

Rys. 5.1. Przykłady wartości współczynników korelacji między fingerprintem wydobytym z pirackiej kopii oraz fingerprintem każdego użytkownika dla metody opartej na iloczynie macierzy (zmówieni piraci mają początkowe wartości identyfikatorów), opracowanie własne.

Rys. 7.1. Szyfrowanie w trybie wiązania kwaternionów zaszyfrowanych CQC w metodzie JFD opartej na rotacji kwaternionów, opracowanie własne.

Rys. 7.2. Deszyfrowanie w trybie wiązania kwaternionów zaszyfrowanych CQC w metodzie JFD opartej na rotacji kwaternionów, opracowanie własne.

Rys. 7.3. Przykłady wartości współczynników korelacji między fingerprintem wydobytym z pirackiej kopii oraz fingerprintem każdego użytkownika dla metody JFD opartej na rotacji kwaternionów (piraci mają początkowe wartości identyfikatorów), opracowanie własne.

Rys. 8.1. Umiejscowienie kwaternionów klucza biorących udział w szyfrowaniu przykładowego kwaternionu obrazu $x_{1,2}$, przy długości serii $Z = 2$, opracowanie własne.

Rys. 8.2. Umiejscowienie kwaternionów klucza biorących udział w deszyfrowaniu przykładowego kwaternionu obrazu $y_{1,2}$, przy długości serii $Z = 2$, opracowanie własne.

Rys. 8.3. Przykłady wartości współczynników korelacji między fingerprintem wydobytym z pirackiej kopii oraz fingerprintem każdego użytkownika dla metody JFD opartej na serii rotacji kwaternionów (piraci mają początkowe wartości identyfikatorów), opracowanie własne.

Rys. 9.1. Metoda A: przykładowe obrazy w postaci a) oryginalnej, b) zaszyfrowanej, c) odszyfrowanej z osadzonym fingerprintem oraz d) fingerprint wydobyty z obrazu odszyfrowanego, dla szyfrowania jednego współczynnika DCT w każdym bloku, opracowanie własne.

Rys. 9.2. Metoda A: przykładowe obrazy w postaci a) oryginalnej, b) zaszyfrowanej, c) odszyfrowanej z osadzonym fingerprintem oraz d) fingerprint wydobyty z obrazu odszyfrowanego, dla szyfrowania jednego współczynnika DCT w każdym bloku, opracowanie własne.

Rys. 9.3. Metoda A: przykładowe obrazy w postaci a) oryginalnej, b) zaszyfrowanej, c) odszyfrowanej z osadzonym fingerprintem oraz d) fingerprint wydobyty z obrazu odszyfrowanego, dla szyfrowania 16 współczynników DCT w każdym bloku, opracowanie własne.

Rys. 9.4. Metoda A: przykładowe obrazy w postaci a) oryginalnej, b) zaszyfrowanej, c) odszyfrowanej z osadzonym fingerprintem oraz d) fingerprint wydobyty z obrazu odszyfrowanego, dla szyfrowania 16 współczynników DCT w każdym bloku, opracowanie własne.

Rys. 9.5. Metoda B: przykładowe obrazy w postaci a) oryginalnej, b) zaszyfrowanej, c) odszyfrowanej z osadzonym fingerprintem oraz d) fingerprint wydobyty z obrazu odszyfrowanego, dla szyfrowania jednego współczynnika DCT w każdym bloku, opracowanie własne.

Rys. 9.6. Metoda B: przykładowe obrazy w postaci a) oryginalnej, b) zaszyfrowanej, c) odszyfrowanej z osadzonym fingerprintem oraz d) fingerprint wydobyty z obrazu odszyfrowanego, dla szyfrowania jednego współczynnika DCT w każdym bloku, opracowanie własne.

Rys. 9.7. Metoda B: przykładowe obrazy w postaci a) oryginalnej, b) zaszyfrowanej, c) odszyfrowanej z osadzonym fingerprintem oraz d) fingerprint wydobyty z obrazu odszyfrowanego, dla szyfrowania 16 współczynników DCT w każdym bloku, opracowanie własne.

Rys. 9.8. Metoda B: przykładowe obrazy w postaci a) oryginalnej, b) zaszyfrowanej, c) odszyfrowanej z osadzonym fingerprintem oraz d) fingerprint wydobyty z obrazu odszyfrowanego, dla szyfrowania 16 współczynników DCT w każdym bloku, opracowanie własne.

Rys. 9.9. Metoda C: przykładowe obrazy w postaci a) oryginalnej, b) zaszyfrowanej, c) odszyfrowanej z osadzonym fingerprintem oraz d) fingerprint wydobyty z obrazu odszyfrowanego, dla szyfrowania jednego współczynnika DCT w każdym bloku, opracowanie własne.

Rys. 9.10. Metoda C: przykładowe obrazy w postaci a) oryginalnej, b) zaszyfrowanej, c) odszyfrowanej z osadzonym fingerprintem oraz d) fingerprint wydobyty z obrazu odszyfrowanego, dla szyfrowania jednego współczynnika DCT w każdym bloku, opracowanie własne.

Rys. 9.11. Metoda C: przykładowe obrazy w postaci a) oryginalnej, b) zaszyfrowanej, c) odszyfrowanej z osadzonym fingerprintem oraz d) fingerprint wydobyty z obrazu odszyfrowanego, dla szyfrowania 16 współczynników DCT w każdym bloku, opracowanie własne.

Rys. 9.12. Metoda C: przykładowe obrazy w postaci a) oryginalnej, b) zaszyfrowanej, c) odszyfrowanej z osadzonym fingerprintem oraz d) fingerprint wydobyty z obrazu odszyfrowanego, dla szyfrowania 16 współczynników DCT w każdym bloku, opracowanie własne.

Rys. 9.13. Metoda A: identyfikacja pirata w przypadku ataku poprzez dodanie szumu dla SNR równego 30 dB, 25 dB, 20 dB, 15 dB, 10 dB oraz 5 dB, opracowanie własne.

Rys. 9.14. Metoda B: identyfikacja pirata w przypadku ataku poprzez dodanie szumu dla SNR równego 30 dB, 25 dB, 20 dB, 15 dB, 10 dB oraz 5 dB, opracowanie własne.

Rys. 9.15. Metoda C: identyfikacja pirata w przypadku ataku poprzez dodanie szumu dla SNR równego 30 dB, 25 dB, 20 dB, 15 dB, 10 dB oraz 5 dB, opracowanie własne.

Rys. 9.16. Metoda A: przykłady identyfikacji pirata w przypadku ataku poprzez kompresję dla 70 %, 50%, 30% oraz 10% jakości obrazu skompresowanego, opracowanie własne.

Rys. 9.17. Metoda B: przykłady identyfikacji pirata w przypadku ataku poprzez kompresję dla 70 %, 50%, 30% oraz 10% jakości obrazu skompresowanego, opracowanie własne.

Rys. 9.18. Metoda C: przykłady identyfikacji pirata w przypadku ataku poprzez kompresję dla 70 %, 50%, 30% oraz 10% jakości obrazu skompresowanego, opracowanie własne.

Rys. 9.19. Metoda A: odporność na ataki zmowy (atak uśredniający, atak minimum, atak maksimum) w przypadku zmowy 20, 40, 60, 80 i 100 piratów, opracowanie własne.

Rys. 9.20. Metoda A: odporność na ataki zmowy (atak medianowy, atak min-max, losowy atak różnicowy) w przypadku zmowy 20, 40, 60, 80 i 100 piratów, opracowanie własne.

Rys. 9.21. Metoda B: odporność na ataki zmowy (atak uśredniający, atak minimum, atak maksimum) w przypadku zmowy 20, 40, 60, 80 i 100 piratów, opracowanie własne.

Rys. 9.22. Metoda B: odporność na ataki zmowy (atak medianowy, atak min-max, losowy atak różnicowy) w przypadku zmowy 20, 40, 60, 80 i 100 piratów, opracowanie własne.

Rys. 9.23. Metoda C: odporność na ataki zmowy (atak uśredniający, atak minimum, atak maksimum) w przypadku zmowy 20, 40, 60, 80 i 100 piratów, przy długości serii $Z = 2$, opracowanie własne.

Rys. 9.24. Metoda C: odporność na ataki zmowy (atak medianowy, atak min-max, losowy atak różnicowy) w przypadku zmowy 20, 40, 60, 80 i 100 piratów, przy długości serii $Z = 2$, opracowanie własne.

Rys. 9.25. Metoda A: odporność na atak zmowy o liczbie 20, 40 ,60, 80 i 100 piratów połączony z kompresją zachowującą 90%, 80%, 70% i 60% jakości obrazu, opracowanie własne.

Rys. 9.26. Metoda B: odporność na atak zmowy o liczbie 20, 40 ,60, 80 i 100 piratów połączony z kompresją zachowującą 90%, 80%, 70% i 60% jakości obrazu, opracowanie własne.

Rys. 9.27. Metoda C: odporność na atak zmowy o liczbie 20, 40 ,60, 80 i 100 piratów połączony z kompresją zachowującą 90%, 80%, 70% i 60% jakości obrazu, dla długości serii $Z = 2$, opracowanie własne.

Rys. 10.1. Ogólny schemat etapu rejestracji anonimowych klientów, opracowanie własne.

Rys. 10.2. Szczegółowy schemat rejestracji anonimowych klientów, opracowanie własne.

Rys. 10.3. Ogólny schemat etapu dystrybucji kluczy deszyfrujących, opracowanie własne.

Rys. 10.4. Szczegółowy schemat dystrybucji kluczy deszyfrujących, opracowanie własne.

Rys. 10.5. Ogólny schemat etapu identyfikacji piratów, opracowanie własne.

Rys. 10.6. Szczegółowy schemat etapu identyfikacji piratów, opracowanie własne.

Tab. 2.1. Priorytety odzwierciedlające istotność poszczególnych wymagań dla różnych zastosowań cyfrowego znakowania danych multimedialnych.

Tab. 9.1. Porównanie nowych metod JFD przedstawionych w rozprawie ze sobą oraz z istniejącymi metodami znanymi z literatury.

ZAŁĄCZNIKI

Załącznik A. Wyprowadzenie przybliżonego wzoru na postać właściwego fingerprintu w metodzie opartej na rotacji kwaternionów

Wszystkie oznaczenia zastosowane w tym załączniku są tożsame z oznaczeniami w rozdziale 7. Właściwa postać fingerprintu $\mathbf{f}_{\text{emb}}^{(u)}$ osadzonego w kwaternionach obrazu *u*-tego użytkownika $\mathbf{x}^{(u)}$ jest zależna w silnie nieliniowy sposób od wektora kwaternionów fingerprintu $\mathbf{f}^{(u)}$ tego użytkownika, wektora kwaternionów klucza szyfrującego $\mathbf{e}$, wektora kwaternionów obrazu oryginalnego $\mathbf{x}$, a w przypadku trybu wiązania kwaternionów również od wektora kwaternionów zaszyfrowanych $\mathbf{y}$, kwaternionu inicjalizującego y_0, kwaternionu przesunięcia z_1 oraz bazy arytmetyki modularnej z_2. Dla uproszczenia przekształceń, przyjęto ogólny przypadek szyfrowania bez wiązania kwaternionów:

$$y_n = e_l \cdot x_n \cdot e_l^{-1}, \tag{A.1}$$

oraz, odpowiednio, deszyfrowania bez wiązania kwaternionów:

$$x_n^{(u)} = d_l^{(u)^{-1}} \cdot y_n \cdot d_l^{(u)}. \tag{A.2}$$

Ponieważ kwaterniony klucza deszyfrującego $d_l^{(u)}$ opisane są równaniem:

$$d_l^{(u)} = e_l + \alpha \cdot f_l^{(u)}, \tag{A.3}$$

zatem wykorzystując równania (A.1), (A.2) oraz (A.3) można zapisać, że:

$$x_n^{(u)} = d_l^{(u)^{-1}} \cdot e_l \cdot x_n \cdot e_l^{-1} \cdot d_l^{(u)}, \tag{A.4}$$

$$x_n^{(u)} = (e_l + \alpha \cdot f_l^{(u)})^{-1} \cdot e_l \cdot x_n \cdot e_l^{-1} \cdot (e_l + \alpha \cdot f_l^{(u)}). \tag{A.5}$$

Ponieważ dla dowolnych kwaternionów q_1 i q_2 zachodzi:

$$q_1^{-1} \cdot q_2 = (q_2^{-1} \cdot q_1)^{-1}, \tag{A.6}$$

zatem można zapisać, że:

$$x_n^{(u)} = (1 + \alpha \cdot e_l^{-1} \cdot f_l^{(u)})^{-1} \cdot x_n \cdot (1 + \alpha \cdot e_l^{-1} \cdot f_l^{(u)}). \tag{A.7}$$

Wyrażenie $(1+\alpha\cdot e_l^{-1}\cdot f_l^{(u)})^{-1}$ jest sumą następującego, zbieżnego (warunkiem zbieżności jest $\alpha<<1$) szeregu geometrycznego:

$$1+(-\alpha\cdot e_l^{-1}\cdot f_l^{(u)})+(-\alpha\cdot e_l^{-1}\cdot f_l^{(u)})^2+\ldots=\sum_{h=0}^{\infty}(-\alpha\cdot e_l^{-1}\cdot f_l^{(u)})^h=(1+\alpha\cdot e_l^{-1}\cdot f_l^{(u)})^{-1} \quad \text{(A.8)}$$

Ponieważ $\alpha<<1$, suma powyższego szeregu geometrycznego może być przybliżona z wykorzystaniem tylko dwóch pierwszych elementów szeregu:

$$(1+\alpha\cdot e_l^{-1}\cdot f_l^{(u)})^{-1}\cong 1-\alpha\cdot e_l^{-1}\cdot f_l^{(u)}. \quad \text{(A.9)}$$

Stąd, można zapisać, że:

$$x_n^{(u)}\cong(1-\alpha\cdot e_l^{-1}\cdot f_l^{(u)})\cdot x_n\cdot(1+\alpha\cdot e_l^{-1}\cdot f_l^{(u)}), \quad \text{(A.10)}$$

$$x_n^{(u)}\cong(1-\alpha\cdot e_l^{-1}\cdot f_l^{(u)})\cdot(x_n+\alpha\cdot x_n\cdot e_l^{-1}\cdot f_l^{(u)}), \quad \text{(A.11)}$$

$$x_n^{(u)}\cong x_n+\alpha\cdot x_n\cdot e_l^{-1}\cdot f_l^{(u)}-\alpha\cdot e_l^{-1}\cdot f_l^{(u)}\cdot x_n-\alpha^2\cdot e_l^{-1}\cdot f_l^{(u)}\cdot x_n\cdot e_l^{-1}\cdot f_l^{(u)}. \quad \text{(A.12)}$$

Ponieważ $\alpha<<1$ oraz $\alpha^2\cong 0$, równość (A.12) można zapisać w przybliżonej postaci:

$$x_n^{(u)}\cong x_n+\alpha\cdot x_n\cdot e_l^{-1}\cdot f_l^{(u)}-\alpha\cdot e_l^{-1}\cdot f_l^{(u)}\cdot x_n, \quad \text{(A.13)}$$

$$x_n^{(u)}\cong x_n+\alpha\cdot(x_n\cdot e_l^{-1}\cdot f_l^{(u)}-e_l^{-1}\cdot f_l^{(u)}\cdot x_n). \quad \text{(A.14)}$$

Z powyższych przekształceń wynika, że właściwa postać fingerprintu $\mathbf{f}_{\text{emb}}^{(u)}$ osadzonego z siłą α w kwaternionach obrazu *u*-tego użytkownika $\mathbf{x}^{(u)}$ jest następująca:

$$f_{n_{\text{emb}}}^{(u)}\cong x_n\cdot e_l^{-1}\cdot f_l^{(u)}-e_l^{-1}\cdot f_l^{(u)}\cdot x_n, \quad \text{(A.15)}$$

$$\mathbf{f}_{\text{emb}}^{(u)}=\left[f_{1_{\text{emb}}}^{(u)}\quad f_{2_{\text{emb}}}^{(u)}\quad\cdots\quad f_{N_{\text{emb}}}^{(u)}\right], \quad \text{(A.16)}$$

gdzie $\mathbf{f}_{\text{emb}}^{(u)}$ jest wektorem kwaternionów, zależnym w nieliniowy sposób od wektora kwaternionów fingerprintu $\mathbf{f}^{(u)}$, wektora kwaternionów klucza szyfrującego $\mathbf{e}$, oraz wektora kwaternionów obrazu oryginalnego $\mathbf{x}$.

Powyższa postać osadzonego fingerprintu jest słuszna również w przypadku szyfrowania z wiązaniem bloków, gdyż dla szyfrowania z wiązaniem kwaternionów:

$$y_n=e_l\cdot[(x_n+z_1+y_{n-1})\bmod z_2]\cdot e_l^{-1}, \quad \text{(A.17)}$$

oraz, odpowiednio, dla deszyfrowania z wiązaniem kwaternionów:

$$x_n^{(u)} = [(d_l^{(u)^{-1}} \cdot y_n \cdot d_l^{(u)} - y_{n-1}) \bmod z_2] - z_1, \tag{A.18}$$

równość (A.14) przyjmuje postać:

$$\begin{aligned} x_n^{(u)} \cong\ & [(x_n + z_1 + y_{n-1}) \bmod z_2] + \\ & + \alpha \cdot \{[(x_n + z_1 + y_{n-1}) \bmod z_2] \cdot e_l^{-1} \cdot f_l^{(u)} - \\ & - e_l^{-1} \cdot f_l^{(u)} \cdot [(x_n + z_1 + y_{n-1}) \bmod z_2] - y_{n-1}\} \bmod z_2 - z_1 \end{aligned}, \tag{A.19}$$

$$\begin{aligned} (x_n^{(u)} + z_1 + y_{n-1}) \bmod z_2 \cong\ & [(x_n + z_1 + y_{n-1}) \bmod z_2] + \\ & + \alpha \cdot \{[(x_n + z_1 + y_{n-1}) \bmod z_2] \cdot e_l^{-1} \cdot f_l^{(u)} - \\ & - e_l^{-1} \cdot f_l^{(u)} \cdot [(x_n + z_1 + y_{n-1}) \bmod z_2]\} \end{aligned}. \tag{A.20}$$

Załącznik B. Wyprowadzenie przybliżonego wzoru na postać właściwego fingerprintu w metodzie opartej na serii rotacji kwaternionów

Wszystkie oznaczenia zastosowane w tym załączniku są tożsame z oznaczeniami w rozdziale 8. Właściwa postać fingerprintu $\mathbf{F}_{\text{emb}}^{(u)}$ osadzonego w kwaternionach obrazu *u*-tego użytkownika $\mathbf{X}^{(u)}$ jest zależna w silnie nieliniowy sposób od macierzy kwaternionów fingerprintu $\mathbf{F}^{(u)}$ tego użytkownika, macierzy kwaternionów klucza szyfrującego $\mathbf{E}$, macierzy kwaternionów obrazu oryginalnego $\mathbf{X}$, a także długości serii rotacji *Z*. Dla przykładu, przyjęto przypadek szyfrowania i deszyfrowania kwaternionów dla długości serii $Z = 2$:

$$y_{n_1,n_2} = e_{g_1(2),g_2(2)} \cdot e_{g_1(1),g_2(1)} \cdot (x_{n_1,n_2} + \beta \cdot \mathrm{im}(e_{h_1,h_2})) \cdot e_{g_1(1),g_2(1)}{}^{*} \cdot e_{g_1(2),g_2(2)}{}^{*}, \tag{B.1}$$

$$x_{n_1,n_2}^{(u)} = d_{g_1(1),g_2(1)}^{(u)}{}^{*} \cdot d_{g_1(2),g_2(2)}^{(u)}{}^{*} \cdot y_{n_1,n_2} \cdot d_{g_1(2),g_2(2)}^{(u)} \cdot d_{g_1(1),g_2(1)}^{(u)} - \beta \cdot \mathrm{im}(d_{h_1,h_2}^{(u)}). \tag{B.2}$$

Ponieważ kwaterniony klucza deszyfrującego $d_{r_1,r_2}^{(u)}$ opisane są równaniem:

$$d_{r_1,r_2}^{(u)} = \frac{e_{r_1,r_2} \cdot (1 + \alpha \cdot f_{r_1,r_2}^{(u)})}{\left\| e_{r_1,r_2} \cdot (1 + \alpha \cdot f_{r_1,r_2}^{(u)}) \right\|}, \tag{B.3}$$

zatem wykorzystując równania (B.1), (B.2) oraz (B.3) można zapisać, że:

$$\begin{aligned} x_{n_1,n_2}^{(u)} = {} & d_{g_1(1),g_2(1)}^{(u)}{}^{*} \cdot d_{g_1(2),g_2(2)}^{(u)}{}^{*} \cdot e_{g_1(2),g_2(2)} \cdot e_{g_1(1),g_2(1)} \cdot (x_{n_1,n_2} + \beta \cdot \mathrm{im}(e_{h_1,h_2})) \cdot \\ & \cdot e_{g_1(1),g_2(1)}{}^{*} \cdot e_{g_1(2),g_2(2)}{}^{*} \cdot d_{g_1(2),g_2(2)}^{(u)} \cdot d_{g_1(1),g_2(1)}^{(u)} - \beta \cdot \mathrm{im}(d_{h_1,h_2}^{(u)}) \end{aligned}, \tag{B.4}$$

$$\begin{aligned} x_{n_1,n_2}^{(u)} = {} & \left(\frac{e_{g_1(1),g_2(1)} \cdot (1 + \alpha \cdot f_{g_1(1),g_2(1)}^{(u)})}{\left\| e_{g_1(1),g_2(1)} \cdot (1 + \alpha \cdot f_{g_1(1),g_2(1)}^{(u)}) \right\|} \right)^{*} \cdot \left(\frac{e_{g_1(2),g_2(2)} \cdot (1 + \alpha \cdot f_{g_1(2),g_2(2)}^{(u)})}{\left\| e_{g_1(2),g_2(2)} \cdot (1 + \alpha \cdot f_{g_1(2),g_2(2)}^{(u)}) \right\|} \right)^{*} \cdot e_{g_1(2),g_2(2)} \cdot e_{g_1(1),g_2(1)} \cdot \\ & \cdot (x_{n_1,n_2} + \beta \cdot \mathrm{im}(e_{h_1,h_2})) \cdot e_{g_1(1),g_2(1)}{}^{*} \cdot e_{g_1(2),g_2(2)}{}^{*} \cdot \left(\frac{e_{g_1(2),g_2(2)} \cdot (1 + \alpha \cdot f_{g_1(2),g_2(2)}^{(u)})}{\left\| e_{g_1(2),g_2(2)} \cdot (1 + \alpha \cdot f_{g_1(2),g_2(2)}^{(u)}) \right\|} \right) \cdot \\ & \cdot \left(\frac{e_{g_1(1),g_2(1)} \cdot (1 + \alpha \cdot f_{g_1(1),g_2(1)}^{(u)})}{\left\| e_{g_1(1),g_2(1)} \cdot (1 + \alpha \cdot f_{g_1(1),g_2(1)}^{(u)}) \right\|} \right) - \beta \cdot \mathrm{im} \left(\frac{e_{h_1,h_2} \cdot (1 + \alpha \cdot f_{h_1,h_2}^{(u)})}{\left\| e_{h_1,h_2} \cdot (1 + \alpha \cdot f_{h_1,h_2}^{(u)}) \right\|} \right). \end{aligned} \tag{B.5}$$

Ponieważ wszystkie kwaterniony klucza szyfrującego e_{r_1,r_2} są znormalizowane, a $\alpha << 1$, to:

$$\frac{1}{\left\| e_{r_1,r_2} \cdot (1 + \alpha \cdot f_{r_1,r_2}^{(u)}) \right\|} \approx 1, \tag{B.6}$$

a w rezultacie, równanie (B.5) upraszcza się do postaci:

$$x_{n_1,n_2}^{(u)} = (e_{g_1(1),g_2(1)} \cdot (1+\alpha \cdot f_{g_1(1),g_2(1)}^{(u)}))^* \cdot (e_{g_1(2),g_2(2)} \cdot (1+\alpha \cdot f_{g_1(2),g_2(2)}^{(u)}))^* \cdot e_{g_1(2),g_2(2)} \cdot e_{g_1(1),g_2(1)} \cdot$$
$$\cdot (x_{n_1,n_2} + \beta \cdot \mathrm{im}(e_{h_1,h_2})) \cdot e_{g_1(1),g_2(1)}^* \cdot e_{g_1(2),g_2(2)}^* \cdot (e_{g_1(2),g_2(2)} \cdot (1+\alpha \cdot f_{g_1(2),g_2(2)}^{(u)})) \cdot$$
$$\cdot (e_{g_1(1),g_2(1)} \cdot (1+\alpha \cdot f_{g_1(1),g_2(1)}^{(u)})) - \beta \cdot \mathrm{im}(e_{h_1,h_2} \cdot (1+\alpha \cdot f_{h_1,h_2}^{(u)})). \tag{B.7}$$

Ponieważ dla dowolnych kwaternionów q_1 i q_2 zachodzi:

$$(q_1 \cdot q_2)^* = q_2^* \cdot q_1^*, \tag{B.8}$$

zatem można zapisać, że:

$$x_{n_1,n_2}^{(u)} = (1+\alpha \cdot f_{g_1(1),g_2(1)}^{(u)})^* \cdot e_{g_1(1),g_2(1)}^* \cdot (1+\alpha \cdot f_{g_1(2),g_2(2)}^{(u)})^* \cdot e_{g_1(2),g_2(2)}^* \cdot e_{g_1(2),g_2(2)} \cdot e_{g_1(1),g_2(1)} \cdot$$
$$\cdot (x_{n_1,n_2} + \beta \cdot \mathrm{im}(e_{h_1,h_2})) \cdot e_{g_1(1),g_2(1)}^* \cdot e_{g_1(2),g_2(2)}^* \cdot e_{g_1(2),g_2(2)} \cdot (1+\alpha \cdot f_{g_1(2),g_2(2)}^{(u)}) \cdot$$
$$\cdot e_{g_1(1),g_2(1)} \cdot (1+\alpha \cdot f_{g_1(1),g_2(1)}^{(u)}) - \beta \cdot \mathrm{im}(e_{h_1,h_2} \cdot (1+\alpha \cdot f_{h_1,h_2}^{(u)})). \tag{B.9}$$

Następnie, ponieważ dla znormalizowanych kwaternionów e_{r_1,r_2} zachodzi:

$$e_{r_1,r_2}^* \cdot e_{r_1,r_2} = e_{r_1,r_2} \cdot e_{r_1,r_2}^* = 1, \tag{B.10}$$

a dla kwaternionów fingerprintu $f_{r_1,r_2}^{(u)}$ zachodzi:

$$(1+\alpha \cdot f_{r_1,r_2}^{(u)})^* = (1+\alpha \cdot f_{r_1,r_2}^{(u)*}), \tag{B.11}$$

wówczas, równanie (B.9) można przekształcić w następujący sposób:

$$x_{n_1,n_2}^{(u)} = (1+\alpha \cdot f_{g_1(1),g_2(1)}^{(u)*}) \cdot e_{g_1(1),g_2(1)}^* \cdot (1+\alpha \cdot f_{g_1(2),g_2(2)}^{(u)*}) \cdot e_{g_1(1),g_2(1)} \cdot$$
$$\cdot (x_{n_1,n_2} + \beta \cdot \mathrm{im}(e_{h_1,h_2})) \cdot e_{g_1(1),g_2(1)}^* \cdot (1+\alpha \cdot f_{g_1(2),g_2(2)}^{(u)}) \cdot$$
$$\cdot e_{g_1(1),g_2(1)} \cdot (1+\alpha \cdot f_{g_1(1),g_2(1)}^{(u)}) - \beta \cdot \mathrm{im}(e_{h_1,h_2} \cdot (1+\alpha \cdot f_{h_1,h_2}^{(u)})), \tag{B.12}$$

$$x_{n_1,n_2}^{(u)} = (e_{g_1(1),g_2(1)}^* \cdot e_{g_1(1),g_2(1)} + \alpha \cdot e_{g_1(1),g_2(1)}^* \cdot f_{g_1(2),g_2(2)}^{(u)*} \cdot e_{g_1(1),g_2(1)} + \alpha \cdot f_{g_1(1),g_2(1)}^{(u)*} \cdot e_{g_1(1),g_2(1)}^* \cdot e_{g_1(1),g_2(1)} +$$
$$+ \alpha^2 \cdot f_{g_1(1),g_2(1)}^{(u)*} \cdot e_{g_1(1),g_2(1)}^* \cdot f_{g_1(2),g_2(2)}^{(u)*} \cdot e_{g_1(1),g_2(1)}) \cdot (x_{n_1,n_2} + \beta \cdot \mathrm{im}(e_{h_1,h_2})) \cdot (e_{g_1(1),g_2(1)}^* \cdot e_{g_1(1),g_2(1)} +$$
$$+ \alpha \cdot e_{g_1(1),g_2(1)}^* \cdot e_{g_1(1),g_2(1)}^* \cdot f_{g_1(1),g_2(1)}^{(u)} + \alpha \cdot e_{g_1(1),g_2(1)}^* \cdot f_{g_1(2),g_2(2)}^{(u)} \cdot e_{g_1(1),g_2(1)} +$$
$$+ \alpha^2 \cdot e_{g_1(1),g_2(1)}^* \cdot f_{g_1(2),g_2(2)}^{(u)} \cdot e_{g_1(1),g_2(1)} \cdot f_{g_1(1),g_2(1)}^{(u)}) - \beta \cdot \mathrm{im}(e_{h_1,h_2} \cdot (1+\alpha \cdot f_{h_1,h_2}^{(u)})), \tag{B.13}$$

$$\begin{aligned} x_{n_1,n_2}^{(u)} &= (1+\alpha\cdot f_{g_1(1),g_2(1)}^{(u)} *+\alpha\cdot e_{g_1(1),g_2(1)} *\cdot f_{g_1(2),g_2(2)}^{(u)} *\cdot e_{g_1(1),g_2(1)} + \\ &+\alpha^2\cdot f_{g_1(1),g_2(1)}^{(u)} *\cdot e_{g_1(1),g_2(1)} *\cdot f_{g_1(2),g_2(2)}^{(u)} *\cdot e_{g_1(1),g_2(1)})\cdot(x_{n_1,n_2}+\beta\cdot \mathrm{im}(e_{h_1,h_2}))\cdot \\ &\cdot(1+\alpha\cdot f_{g_1(1),g_2(1)}^{(u)} +\alpha\cdot e_{g_1(1),g_2(1)} *\cdot f_{g_1(2),g_2(2)}^{(u)} \cdot e_{g_1(1),g_2(1)} + \\ &+\alpha^2\cdot e_{g_1(1),g_2(1)} *\cdot f_{g_1(2),g_2(2)}^{(u)} \cdot e_{g_1(1),g_2(1)}\cdot f_{g_1(1),g_2(1)}^{(u)})-\beta\cdot \mathrm{im}(e_{h_1,h_2}\cdot(1+\alpha\cdot f_{h_1,h_2}^{(u)})), \end{aligned} \tag{B.14}$$

stąd po przemnożeniu wszystkich składników iloczynu, oraz przy założeniu, że $\alpha^2 \cong \alpha^3 \cong \alpha^4 \cong 0$ ponieważ $\alpha << 1$, równość przyjmuje uproszczoną postać:

$$\begin{aligned} x_{n_1,n_2}^{(u)} &\cong (x_{n_1,n_2}+\beta\cdot \mathrm{im}(e_{h_1,h_2}))+ \\ &+\alpha\cdot f_{g_1(1),g_2(1)}^{(u)} *\cdot(x_{n_1,n_2}+\beta\cdot \mathrm{im}(e_{h_1,h_2}))+ \\ &+\alpha\cdot(x_{n_1,n_2}+\beta\cdot \mathrm{im}(e_{h_1,h_2}))\cdot f_{g_1(1),g_2(1)}^{(u)} + \\ &+\alpha\cdot e_{g_1(1),g_2(1)} *\cdot f_{g_1(2),g_2(2)}^{(u)} *\cdot e_{g_1(1),g_2(1)}\cdot(x_{n_1,n_2}+\beta\cdot \mathrm{im}(e_{h_1,h_2}))+ \\ &+\alpha\cdot(x_{n_1,n_2}+\beta\cdot \mathrm{im}(e_{h_1,h_2}))\cdot e_{g_1(1),g_2(1)} *\cdot f_{g_1(2),g_2(2)}^{(u)} \cdot e_{g_1(1),g_2(1)} - \beta\cdot \mathrm{im}(e_{h_1,h_2}\cdot(1+\alpha\cdot f_{h_1,h_2}^{(u)})). \end{aligned} \tag{B.15}$$

Ponieważ zachodzi:

$$\beta\cdot \mathrm{im}(e_{h_1,h_2}\cdot(1+\alpha\cdot f_{h_1,h_2}^{(u)})) = \beta\cdot \mathrm{im}(e_{h_1,h_2}) + \alpha\cdot\beta\cdot \mathrm{im}(e_{h_1,h_2}\cdot f_{h_1,h_2}^{(u)}), \tag{B.16}$$

zatem można zapisać, że:

$$\begin{aligned} x_{n_1,n_2}^{(u)} &\cong x_{n_1,n_2}+\alpha\cdot(f_{g_1(1),g_2(1)}^{(u)} *\cdot(x_{n_1,n_2}+\beta\cdot \mathrm{im}(e_{h_1,h_2}))+(x_{n_1,n_2}+\beta\cdot \mathrm{im}(e_{h_1,h_2}))\cdot f_{g_1(1),g_2(1)}^{(u)} + \\ &+e_{g_1(1),g_2(1)} *\cdot f_{g_1(2),g_2(2)}^{(u)} *\cdot e_{g_1(1),g_2(1)}\cdot(x_{n_1,n_2}+\beta\cdot \mathrm{im}(e_{h_1,h_2}))+ \\ &+(x_{n_1,n_2}+\beta\cdot \mathrm{im}(e_{h_1,h_2}))\cdot e_{g_1(1),g_2(1)} *\cdot f_{g_1(2),g_2(2)}^{(u)} \cdot e_{g_1(1),g_2(1)} - \beta\cdot \mathrm{im}(e_{h_1,h_2}\cdot f_{h_1,h_2}^{(u)})), \end{aligned} \tag{B.17}$$

Z powyższych przekształceń wynika, że właściwa postać fingerprintu $\mathbf{F}_{\mathrm{emb}}^{(u)}$ osadzonego z siłą α w kwaternionach obrazu u-tego użytkownika $\mathbf{X}^{(u)}$ jest następująca:

$$\begin{aligned} f_{r_1,r_2\,\mathrm{emb}}^{(u)} &\cong f_{g_1(1),g_2(1)}^{(u)} *\cdot(x_{n_1,n_2}+\beta\cdot \mathrm{im}(e_{h_1,h_2}))+(x_{n_1,n_2}+\beta\cdot \mathrm{im}(e_{h_1,h_2}))\cdot f_{g_1(1),g_2(1)}^{(u)} + \\ &+e_{g_1(1),g_2(1)} *\cdot f_{g_1(2),g_2(2)}^{(u)} *\cdot e_{g_1(1),g_2(1)}\cdot(x_{n_1,n_2}+\beta\cdot \mathrm{im}(e_{h_1,h_2}))+ \\ &+(x_{n_1,n_2}+\beta\cdot \mathrm{im}(e_{h_1,h_2}))\cdot e_{g_1(1),g_2(1)} *\cdot f_{g_1(2),g_2(2)}^{(u)} \cdot e_{g_1(1),g_2(1)} - \beta\cdot \mathrm{im}(e_{h_1,h_2}\cdot f_{h_1,h_2}^{(u)}), \end{aligned} \tag{B.18}$$

$$\mathbf{F}_{\mathrm{emb}}^{(u)} = \begin{bmatrix} f_{1,1\ \mathrm{emb}}^{(u)} & f_{1,2\ \mathrm{emb}}^{(u)} & \cdots & f_{1,R_2\,\mathrm{emb}}^{(u)} \\ f_{2,1\ \mathrm{emb}}^{(u)} & f_{2,2\ \mathrm{emb}}^{(u)} & \cdots & f_{2,R_2\,\mathrm{emb}}^{(u)} \\ \vdots & \vdots & \ddots & \vdots \\ f_{R_1,1\,\mathrm{emb}}^{(u)} & f_{R_1,2\,\mathrm{emb}}^{(u)} & \cdots & f_{R_1,R_2\,\mathrm{emb}}^{(u)} \end{bmatrix}, \tag{B.19}$$

gdzie $\mathbf{F}_{\text{emb}}^{(u)}$ jest macierzą kwaternionów, zależną w nieliniowy sposób od macierzy kwaternionów fingerprintu $\mathbf{F}^{(u)}$, macierzy kwaternionów klucza szyfrującego $\mathbf{E}$, oraz macierzy kwaternionów obrazu oryginalnego $\mathbf{X}$.

Powyższe przekształcenia można również przeprowadzić dla dowolnej długości serii rotacji *Z* uzyskując ogólną postać równania (B.17):

$$\begin{aligned} x_{n_1,n_2}^{(u)} \cong\ & x_{n_1,n_2} + \alpha \cdot (f_{g_1(1),g_2(1)}^{(u)} * \cdot (x_{n_1,n_2} + \beta \cdot \mathrm{im}(e_{h_1,h_2})) + (x_{n_1,n_2} + \beta \cdot \mathrm{im}(e_{h_1,h_2})) \cdot f_{g_1(1),g_2(1)}^{(u)} + \\ & + e_{g_1(1),g_2(1)} * \cdot f_{g_1(2),g_2(2)}^{(u)} * \cdot e_{g_1(1),g_2(1)} \cdot (x_{n_1,n_2} + \beta \cdot \mathrm{im}(e_{h_1,h_2})) + \\ & + (x_{n_1,n_2} + \beta \cdot \mathrm{im}(e_{h_1,h_2})) \cdot e_{g_1(1),g_2(1)} * \cdot f_{g_1(2),g_2(2)}^{(u)} \cdot e_{g_1(1),g_2(1)} - \beta \cdot \mathrm{im}(e_{h_1,h_2} \cdot f_{h_1,h_2}^{(u)})), \end{aligned} \tag{B.20}$$

a w rezultacie, ogólną postać przybliżonego wzoru na postać właściwego fingerprintu:

$$\begin{aligned} f_{r_1,r_2\,\text{emb}}^{(u)} \cong \sum_{z=1}^{Z} \Bigg[& \left(\prod_{l=1}^{z-1} e_{g_1(l),g_2(l)} * \right) \cdot f_{g_1(z),g_2(z)}^{(u)} * \cdot \left(\prod_{l=1}^{z-1} e_{g_1(z-l),g_2(z-l)} \right) \cdot (x_{n_1,n_2} + \beta \cdot \mathrm{im}(e_{h_1,h_2})) + \\ & + (x_{n_1,n_2} + \beta \cdot \mathrm{im}(e_{h_1,h_2})) \cdot \left(\prod_{l=1}^{z-1} e_{g_1(l),g_2(l)} * \right) \cdot f_{g_1(z),g_2(z)}^{(u)} \cdot \left(\prod_{l=1}^{z-1} e_{g_1(z-l),g_2(z-l)} \right) \Bigg] - \\ & - \beta \cdot \mathrm{im}(e_{h_1,h_2} \cdot f_{h_1,h_2}^{(u)}). \end{aligned} \tag{B.21}$$

Załącznik C. Płyta DVD z kodem źródłowym

Spis plików i folderów na płycie:

- plik spis_plikow.txt – spis plików i folderów na płycie,
- plik Czaplewski_Bartosz_rozprawa.docx – treść rozprawy doktorskiej w formacie docx,
- plik Czaplewski_Bartosz_rozprawa.pdf – treść rozprawy doktorskiej w formacie pdf,
- folder „metoda JFD oparta na iloczynie macierzy" – kod źródłowy metody JFD opartej na iloczynie macierzy, składający się z następujących plików:
  - main.m – funkcja główna uruchamiająca symulację
  - nadawcza.m – funkcja realizująca zadania strony nadawczej
  - odbiorcza.m – funkcja realizująca zadania strony odbiorczej
  - detekcja.m – funkcja realizująca identyfikację piratów
  - atak.m – funkcja realizująca ataki zmowy
  - kompresja.m – funkcja realizująca kompresję JPEG
  - gold.m – funkcja pomocnicza generująca ciągi Golda
  - gzigzag.m – funkcja pomocnicza do wyznaczania kolejności zigzag
  - grayPSNR.m – funkcja pomocnicza do pomiaru PSNR
- folder „metoda JFD oparta na rotacji kwaternionow z CQC" – kod źródłowy metody JFD opartej na rotacji kwaternionów z wiązaniem kwaternionów zaszyfrowanych, składający się z następujących plików:
  - main.m – funkcja główna uruchamiająca symulację
  - nadawcza.m – funkcja realizująca zadania strony nadawczej
  - odbiorcza.m – funkcja realizująca zadania strony odbiorczej
  - detekcja.m – funkcja realizująca identyfikację piratów
  - zmowa.m – funkcja realizująca ataki zmowy
  - kompresja.m – funkcja realizująca kompresję JPEG
  - gold.m – funkcja pomocnicza generująca ciągi Golda
  - gzigzag.m – funkcja pomocnicza do wyznaczania kolejności zigzag
  - colorPSNR.m – funkcja pomocnicza do pomiaru PSNR
- folder „metoda JFD oparta na serii rotacji kwaternionow" – kod źródłowy metody JFD opartej na serii rotacji kwaternionów, składający się z następujących plików:
  - main.m – funkcja główna uruchamiająca symulację
  - nadawcza.m – funkcja realizująca zadania strony nadawczej
  - odbiorcza.m – funkcja realizująca zadania strony odbiorczej
  - detekcja.m – funkcja realizująca identyfikację piratów
  - zmowa.m – funkcja realizująca ataki zmowy
  - kompresja.m – funkcja realizująca kompresję JPEG
  - gold.m – funkcja pomocnicza generująca ciągi Golda
  - gzigzag.m – funkcja pomocnicza do wyznaczania kolejności zigzag
  - colorPSNR.m – funkcja pomocnicza do pomiaru PSNR
- folder „rysunki" – zbiór rysunków wykorzystanych w rozprawie,
- folder „obrazy" – zbiór obrazów wykorzystanych w badaniach.