Collusion Resistant Watermarking Using Convolutional Encoding and Random Spreading

Suggesting: Collusion resistant codes, Error Correcting Codes and watermarking

Abdul Rehman[1,2], Gaetan Le Guelvouit[1], Jean Dion[1]
Frederic Guilloud[2], Matthieu Arzel[2]

IRT b <> com, Cesson Sevigne, France[1]
IMT Atlantique, Lab – STICC, Brest, France[2]
Contact email: **abdul.rehman@b-com.com**

Abdul Rehman

Abdul Rehman earned a master's degree in multimedia networking from Telecom Paris Tech, France in 2020. He is now pursuing a doctorate in cyber security in multimedia at IMT Atlantique. The PhD is funded by IRT b<>com, a private research center in France.

His research focuses on the intersections of multimedia processing, security, and handling.

1. Aims and Contributions of Our Paper

In our paper, we aimed at:

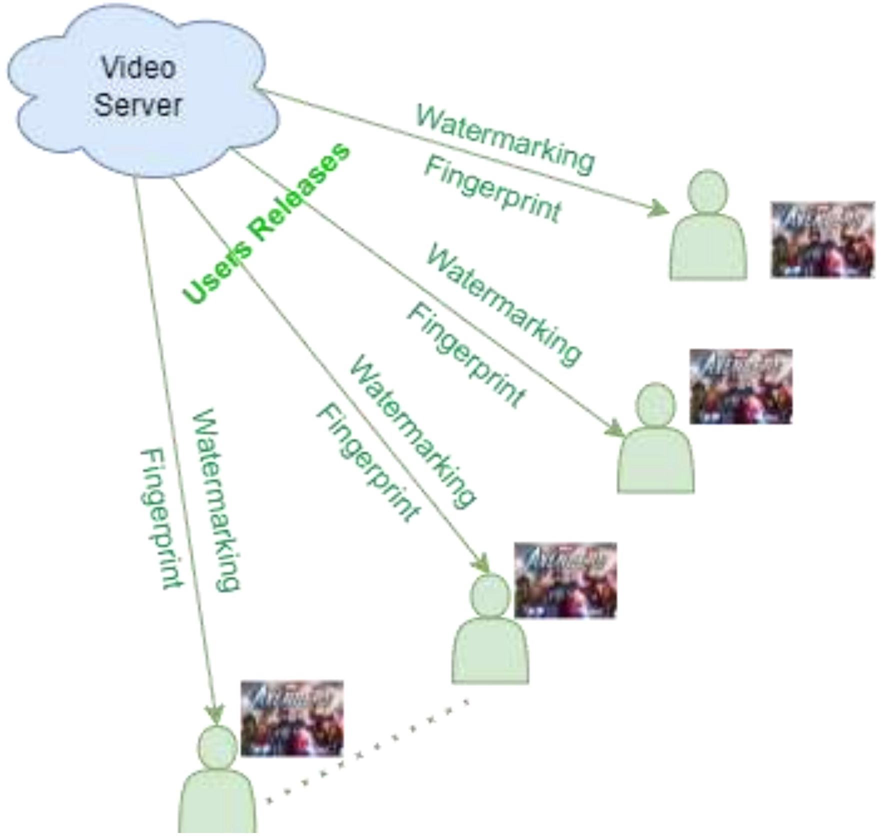
1. Developing an efficient yet secure collusion resistant watermarking method for videos.
2. Comparing the effectiveness of the method in binary and video domains.

Our study makes three important contributions:

1. We suggested employing error-correcting codes (convolutional codes) to reduce overall error rates.
2. Instead of typical convolutional codes, we proposed two different ways of combining spreading with convolutional codes.
3. Then we compared the performance of the two approaches in the binary and video domains.

2.1 The Environment

- Video on demand distribution.
  - A server distributes a video to all users.
  - ***Fingerprint:***
    - A unique code for all the users.
  - ***Watermarking:***
    - A process of embedding a fingerprint into a video.
- Set of authorized users.
  - A limited set of n users receives the video.



- The **Problem:**

 **Threat of suspicious release.**

2.2 The Problem: Threat of Suspicious Release

- **Fingerprints:**
  - Each user has its own fingerprint.
  - With certain length $k\ bits$.
- **Collusion:**
  - A group of users conspiring to deceive the video provider.
  - Creating a suspicious release of video.

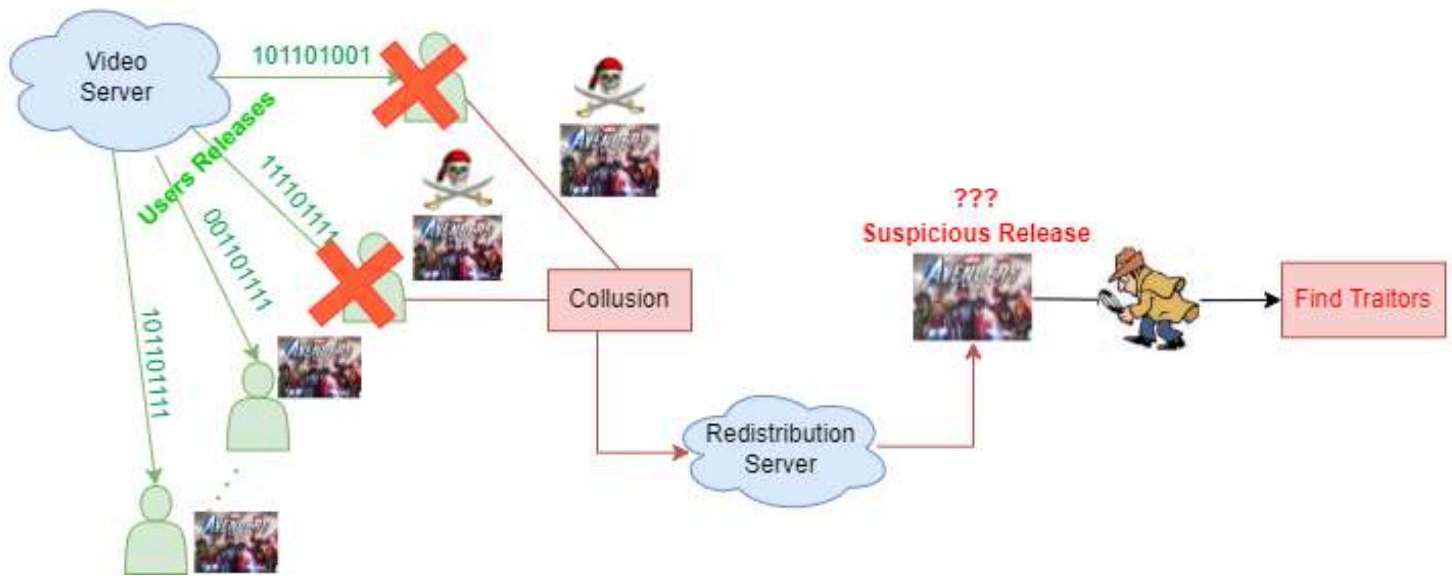
Goal

Catch one

Identify one user who participated in collusion.

Catch all

Identify all users who participated in collusion.



2.3 Collision Attacks Models

- **Collusion attacks models:**
  - Majority vote attack
  - Minority vote attack

- **Detectable positions:**
  - Where they can see the difference between the bits.
  - They can vote to create a suspicious copy.

Solution

Collusion Resistant Fingerprinting Codes

Majority vote attack

"Detectable Positions"

user #1	1	0	1	1	0	1	0	0	1
user #2	1	1	1	1	0	1	1	1	1
user #3	0	0	1	1	0	1	1	1	1
Collusion	1	0	1	1	0	1	1	1	1

???

Minority vote attack

user #1	1	0	1	1	0	1	0	0	1
user #2	1	1	1	1	0	1	1	1	1
user #3	0	0	1	1	0	1	1	1	1
Collusion	0	1	1	1	0	1	0	0	1

???

2.4 Tardos: Collusion Resistant Fingerprinting Codes

Gabor Tardos proposed optimal probabilistic collusion resistant fingerprinting codes in 2003.

- Maximum number of colluders - c_{max}
- False accusation error probability – η
- Number of users – n

Length of codes

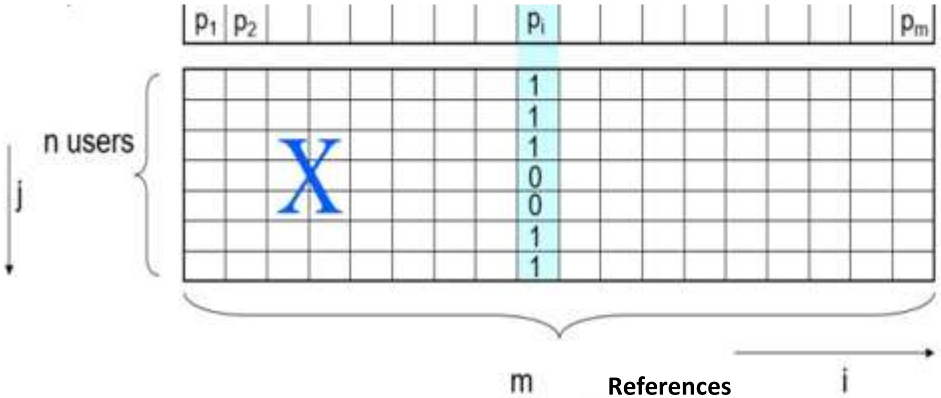
$$k = {c_{max}}^2 \log {}^{n}/_{\eta}$$

How to generate Tardos codes?

- $\forall i \in [1:k]$, Generate $\boldsymbol{p_i}$ in interval $[0,1]$.
  - $\boldsymbol{p_i}$ is i.i.d uniform random variable.
- Generate $\boldsymbol{X}$ matrix of size $n \times k$:
  - Based on Bernoulli($\boldsymbol{p_i}$)
    - Generate code words $\{0,1\}^k$.
  - Save $\boldsymbol{X}$ matrix and $\boldsymbol{p_i}$ vector for decoding.
- **Based on $\boldsymbol{p_i}$ different generation methods for codes:**
  - We used Tardos.

How to decode Tardos codes?

- When a pirated copy y is found:
  - Calculate: $\forall j \in [1:n], \sigma_j = \boldsymbol{g}(\boldsymbol{X}, y_i, p_i)$
- τ is a threshold
- If $\sum_{i=1}^{m} \sigma_j > \tau$, then user j is accused.
- **We used $\boldsymbol{g}(\boldsymbol{X}, \boldsymbol{y_i}, \boldsymbol{p_i})$ function:**
  - Laarhoven



References

1. Gábor Tardos. “Optimal Probabilistic Fingerprint Codes”.In:2003, pp.116–125.URL:https://citeseerx .ist.psu.edu/viewdoc/download?doi=10.1.1.8.8911&rep=rep1&type=pdf.
2. T. Laarhoven and B. de Weger. “Discrete distributions in the tardos scheme, revisited”. In: (2018).URL: arXiv:1302.1741v2[cs.CR]29Apr2013.
3. **A. Rehman, G. Le Guelvouit, J. Dion, F. Guilloud and M. Arzel, "DWT Collusion Resistant Video Watermarking Using Tardos Family Codes," *2022 IEEE 5th International Conference on Image Processing Applications and Systems (IPAS)*, Genova, Italy, 2022, pp. 1-6, doi: 10.1109/IPAS55744.2022.10053023**

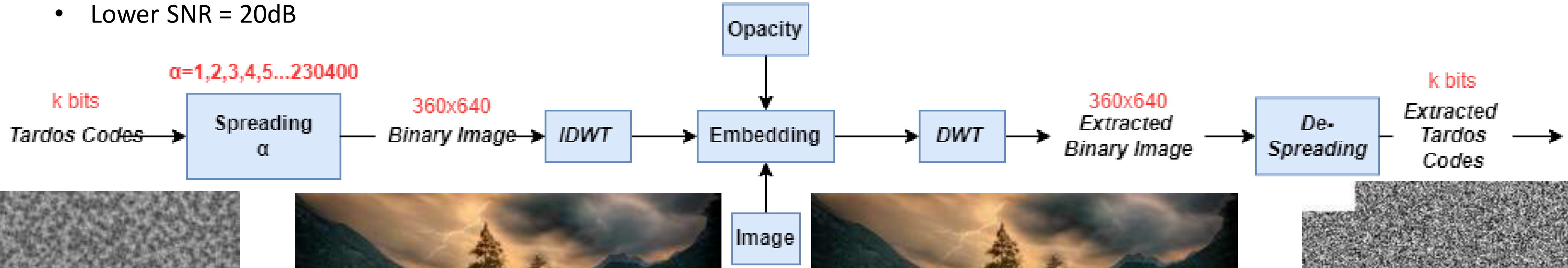
3.1 A Model for Collusion-Resistant Watermarking

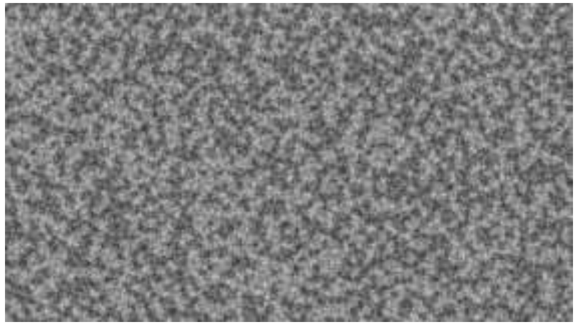
- **$360p$ watermark image:**
  - $(360 \times 640) = 230400 - \{LL3 = (90 \times 40)\} = 226800$
  - $msg\_len = 226800$
- **Alpha Blending:**
  - $I_{wt} = I_i \times opacity + I_w \times (1 - opacity)$
- **Discrete Watermarking (watermark not visible):**
  - Opacity should be close to 1:
    - Lower SNR = 20dB

Solution

Random Spreading

ECC (Error Correcting Codes)

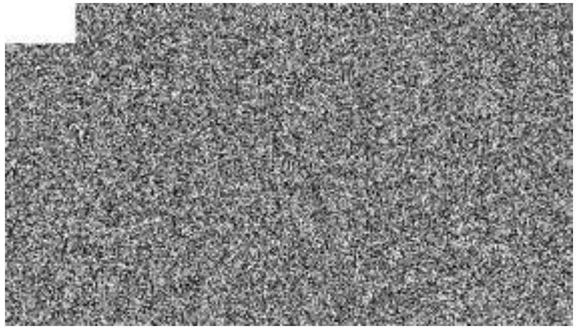




Watermark (I_w)

Original Frame (I_i)

watermarked Frame (I_{wt})



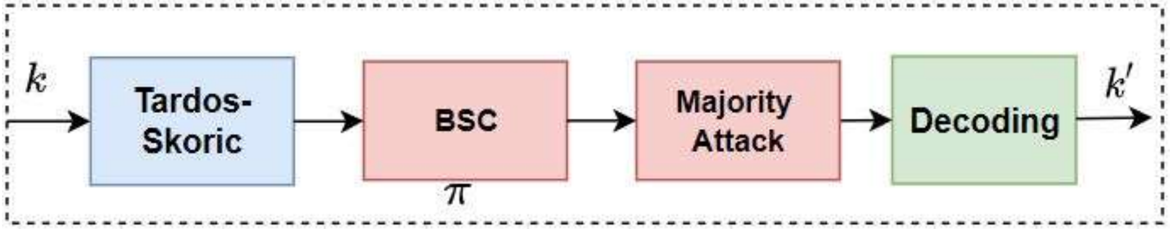
Extracted watermark

3.2 Problem: Tardos Codes vs Errors

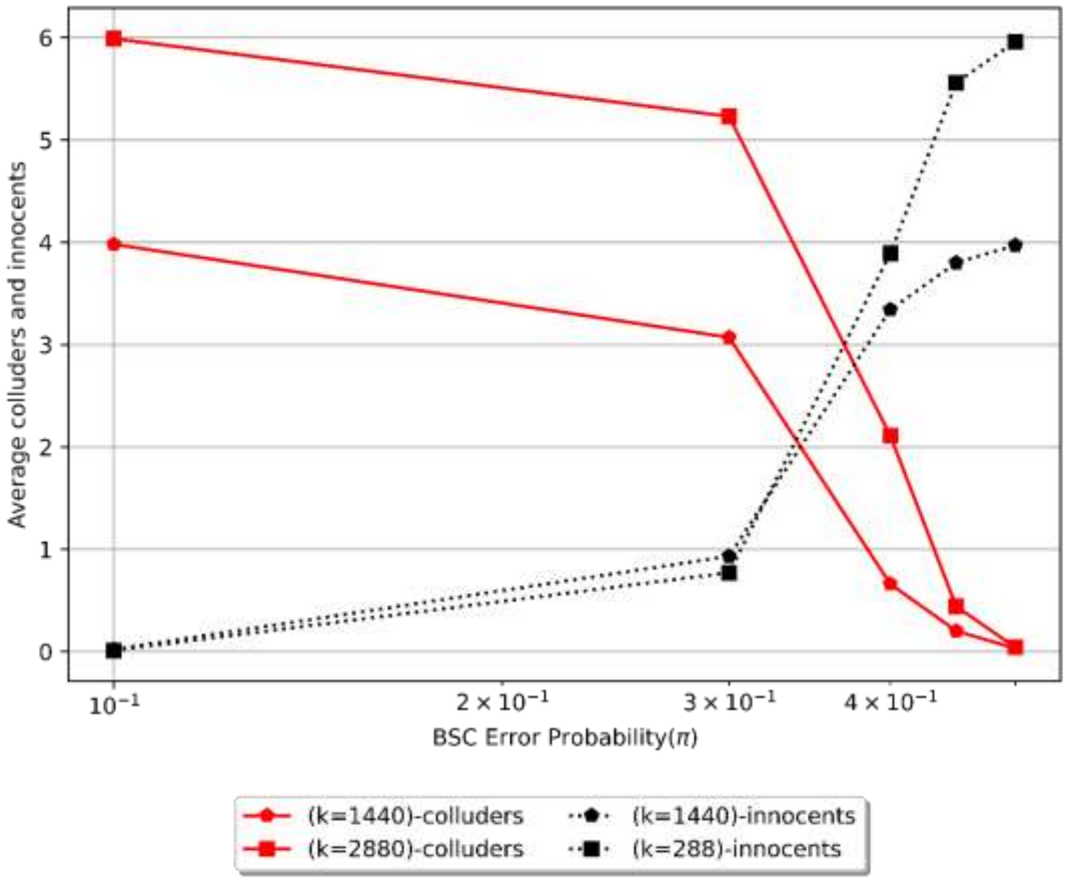
- **Tardos:**

n	η	c_o	k
1000	10^{-3}	4	1440
		6	2880

- **Simulation Parameters**
  - Binary Symmetric Channel (BSC) with $\pi \in [0.1, 0.5]$
  - Random colluders

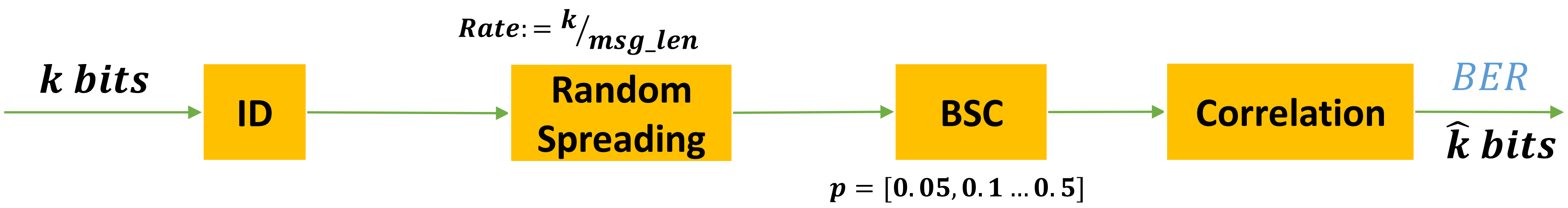


- **Whatever the length of the fingerprint *k*:**
  - Average detected colluders drops:
    - Binary error probabilities higher than $\pi = 2.10^{-1}$



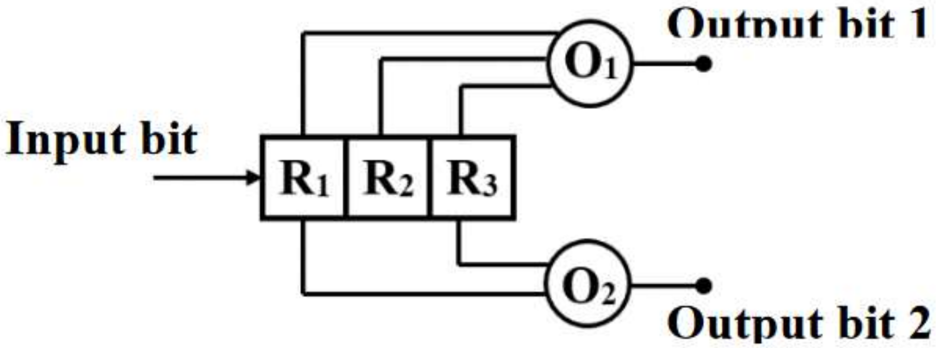
4.1 Random Spreading

- Random Sequence spreading:
  - For each bit in fingerprint:
    - A random sequences is generated r_s.
- For decoding, a correlation is calculated between the received sequence and the original ones.



4.2 Error Correcting Codes (ECC)

- Convolutional Encoder:
  - characterized by three parameters $[k, n, T]$.
    - k: input data length,
    - n: output message length,
    - r: code rate k/n
  - T is the constraint length:
    - which is simply the length of the used register (memory).

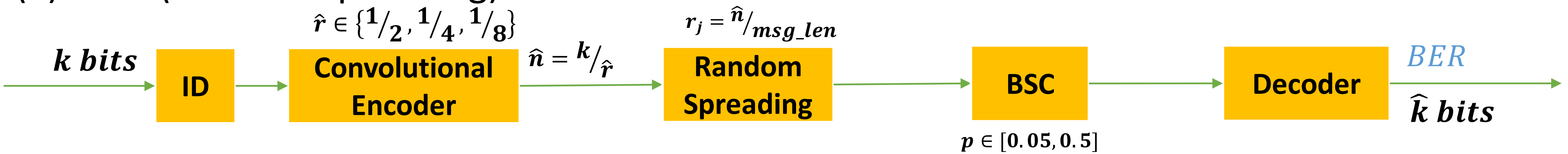


Reference

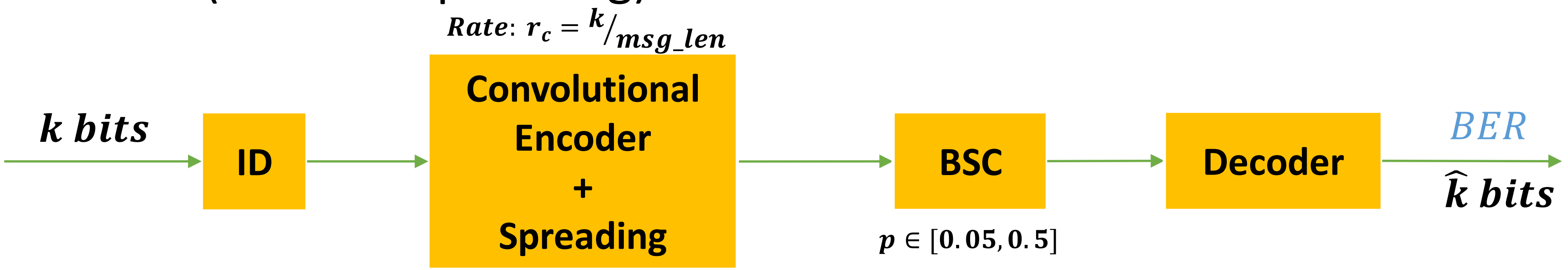
G. Forney, "Convolutional codes I: Algebraic structure," in *IEEE Transactions on Information Theory*, vol. 16, no. 6, pp. 720-738, November 1970, doi: 10.1109/TIT.1970.1054541.

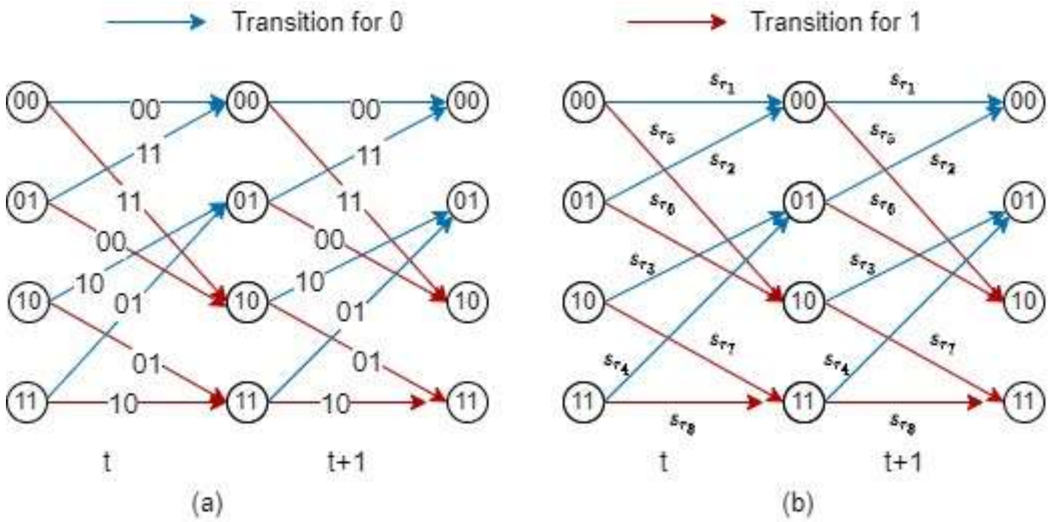
4.2 Coded Approaches: Joint and Concatenated

- (a) Joint (ECC and Spreading):



- (b) Concatenated (ECC and Spreading):





4.3 Uncoded vs Coded Approaches

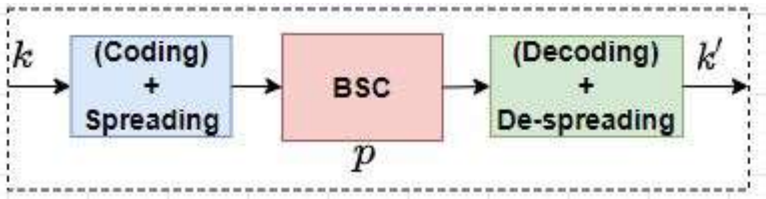
- **Chain settings**
  - Random $\boldsymbol{k} = \mathbf{1440}$
  - $msg\_len = 226800$

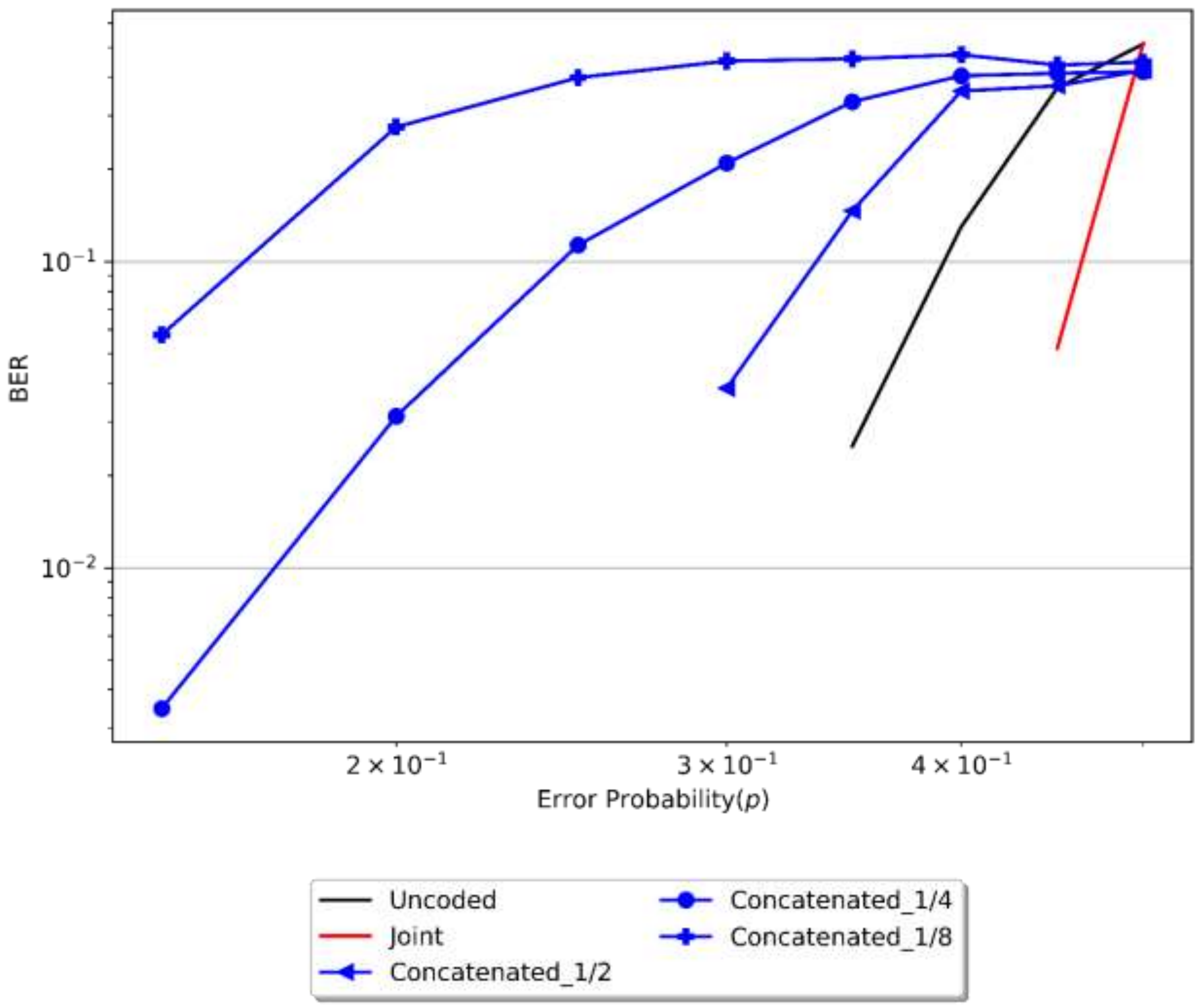
- **Concatenated Coded Approach**:
  - $\boldsymbol{r_c} = {}^{k}/_{msg\_len}$

- **Joint Coded Approach**:
  - $\hat{\boldsymbol{r}} \in \{{}^{1}/_{2}, {}^{1}/_{4}, {}^{1}/_{8}\}$,
  - $\hat{\boldsymbol{n}} = {}^{k}/_{\hat{r}}$,
  - $\boldsymbol{r_j} = {}^{\hat{n}}/_{msg\_len}$

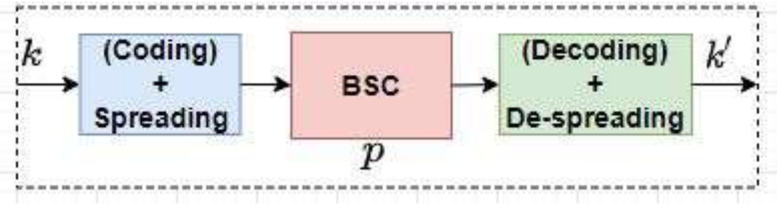
- ***BER*** **?:**
  - *Joint?*
  - Concatenated?
  - *Uncoded?*

Joint coding		
r_c		
$\frac{1440}{226800}$		
Concatenated coding		
$\hat{r}$	$\hat{n}$	r_j
1 (uncoded)	1440	$\frac{1440}{226800}$
$\frac{1}{2}$	2880	$\frac{2880}{226800}$
$\frac{1}{4}$	5760	$\frac{5760}{226800}$
$\frac{1}{8}$	11520	$\frac{11520}{226800}$



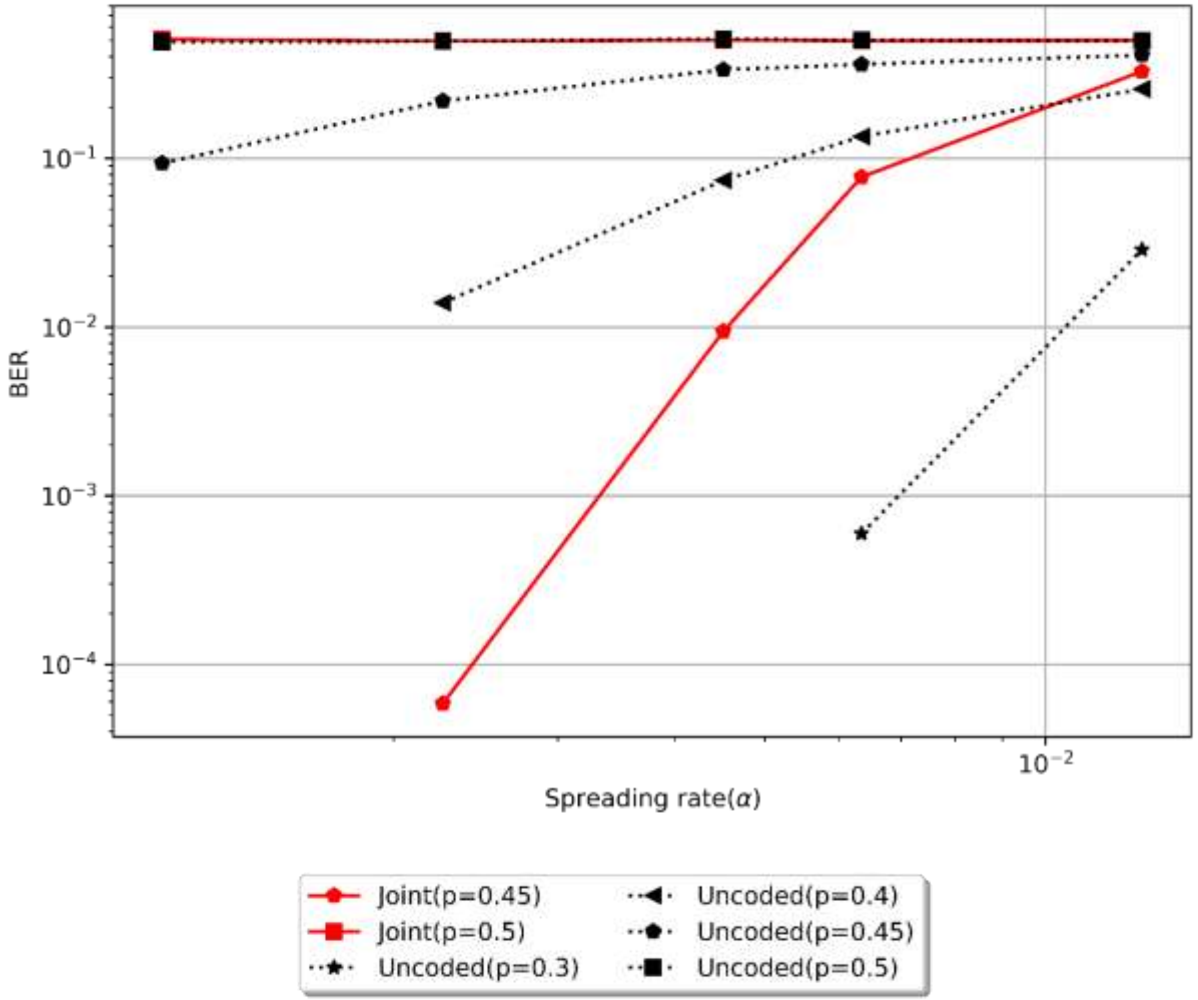


4.5 Spreading Rate vs BER



- To find possible code length with acceptable BER:
  - $k \in \{256, 512, 1024, 1440, 2880\}$
  - $msg\_len = 226800$
- Target:
  - BER = 10^{-2}
  - $p = 0.3$
- Two optimized configurations:

α	k
$\frac{1}{157}$	1440
$\frac{4}{315}$	2880

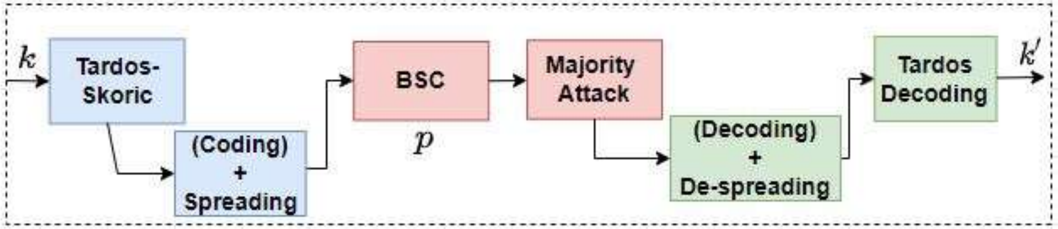


4.6 Binary vs Videos

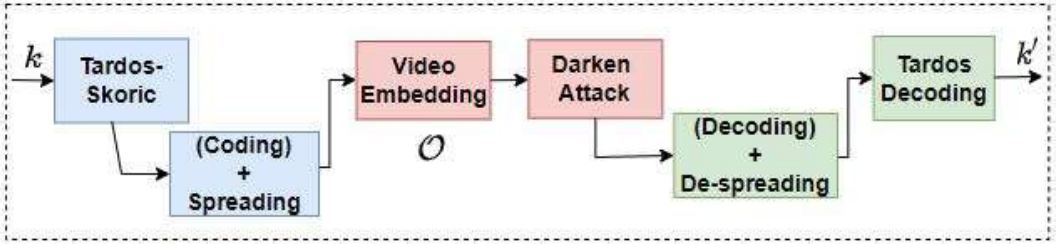
- Tardos settings:

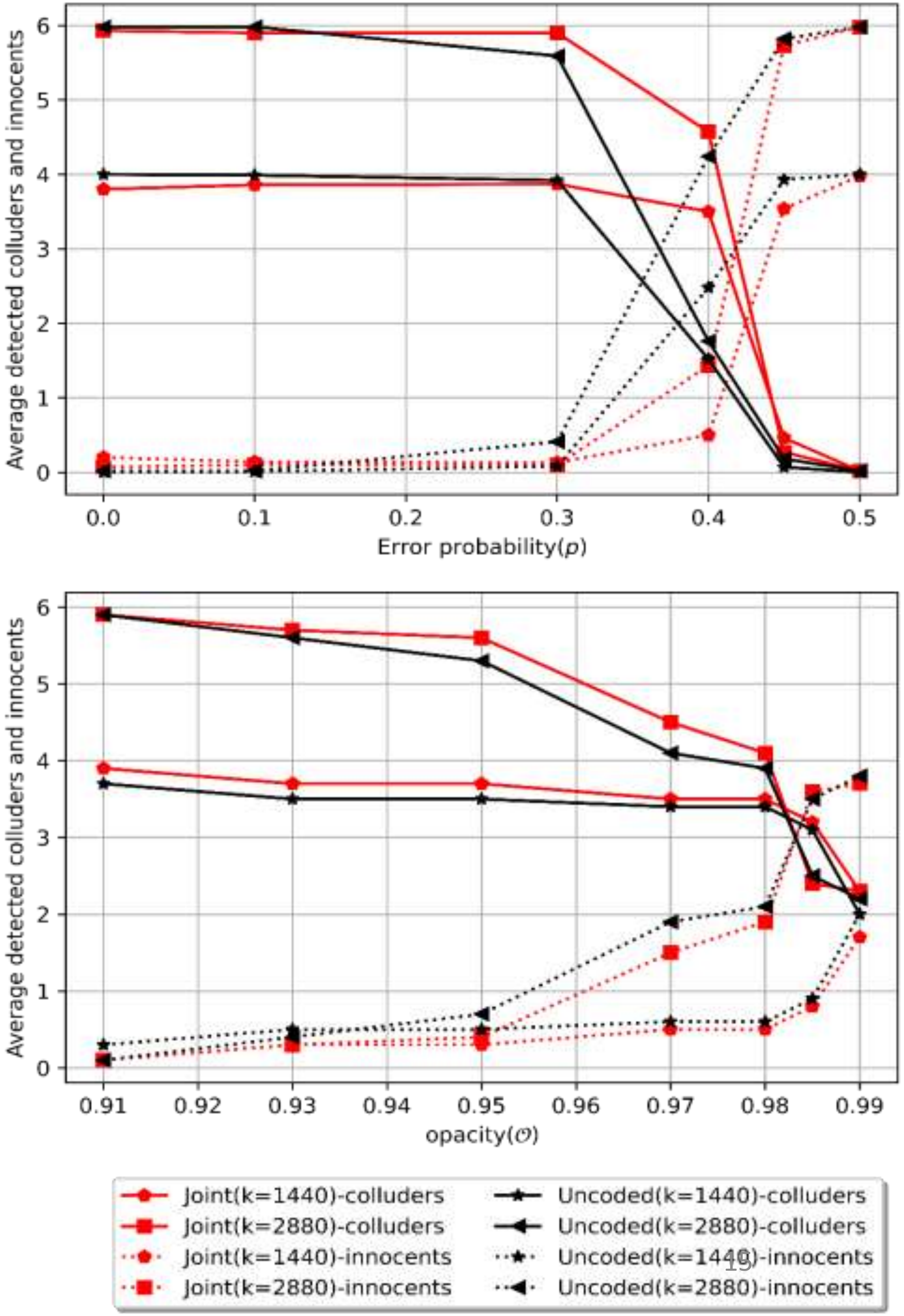
n	η	c_o	k
1000	10^{-3}	4	1440
		6	2880

- Binary Simulation Model:
  - BSC: p ∈ [0.1,0.5]



- Videos Simulation:
  - Opacity ∈ [0.91,0.99]





5. Conclusion & Future Directions

- **Conclusion:**
  - Tardos codes are used to detect traitors.
  - Discrete watermarking using Tardos codes implies extremely low SNR.
  - Spreading improves SNR but reduces Tardos code length.
  - We suggested utilizing convolutional codes with spreading to improve performance.
  - Performance was measured using a 360p watermark image and a 1080p video.
  - The joint coded technique outperforms the un-coded method both theoretically and practically.
- **Future Directions:**
  - More robust error-correcting codes combine with spreading can be investigated for further performance improvement.

Thank you !!
Please don’t hesitate to send an email for questions.

Collusion Resistant Watermarking Using Convolutional Encoding and Random Spreading

Suggesting: Collusion resistant codes, Error Correcting Codes and watermarking

Abdul Rehman[1], Gaetan Le Guelvouit [1], Jean Dion[1]
Frederic Guilloud [2], Matthieu Arzel [2]

IRT b <> com, Cesson Sevigne, France[1]
IMT Atlantique, Lab – STICC, Brest, France[2]
Contact email: **abdul.rehman@b-com.com**

b<>com
IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom
IARIA