Random Distance Method에 대한 제 2 역상 공격

(Second Preimage attack on Random Distance Method)

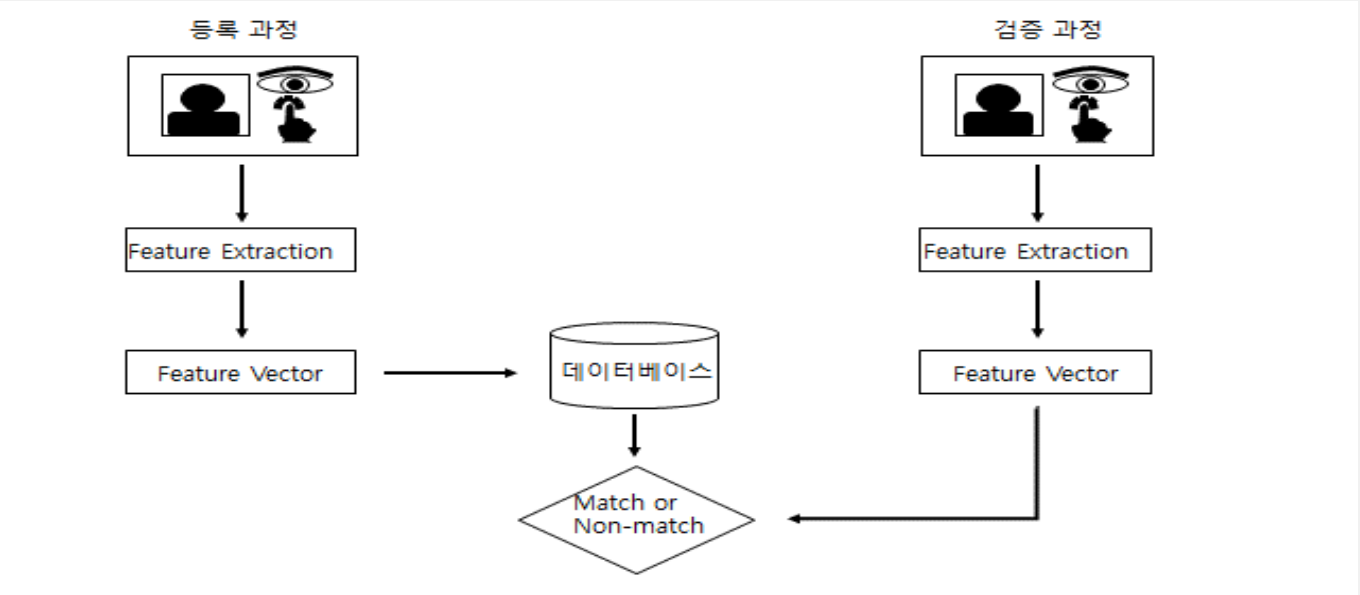
2021.10.16.(토)
전남대학교 대학원
정보보안협동과정
김현석

CONTENTS

01 연구 배경 및 목적

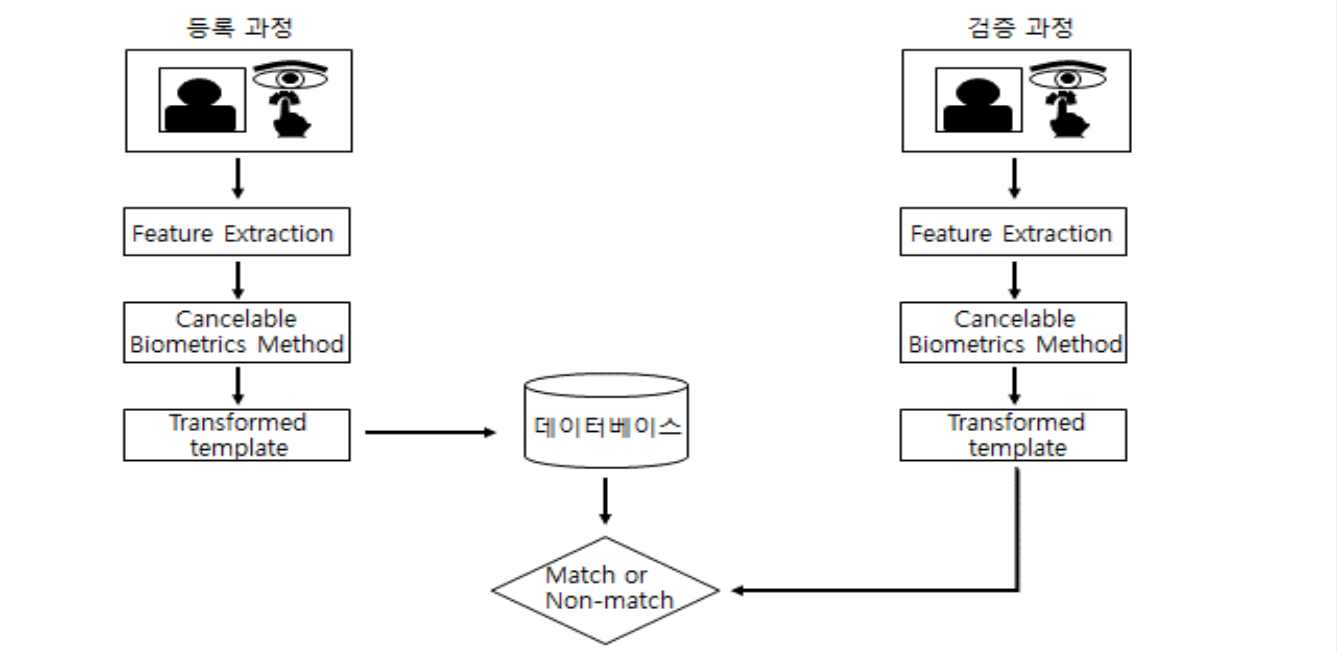
01 연구 배경 및 목적

- 생체 기반 인증은 인증을 위해 사용자가 어떠한 것을 기억하거나 소유할 필요가 없음
- 지식 기반 인증과 소유 기반 인증에 비해 보안성 또한 높은 것으로 여겨짐
- 다양한 시스템 및 산업에서 생체 기반 인증이 채택되고 있는 추세



01 연구 배경 및 목적

- 생체 특징은 쉽게 변하지 않고 인위적으로 변경시키기가 어려움
- 만약 생체 데이터에 대한 유출 및 도난 문제가 발생한다면 이는 심각한 피해로 이어짐
- 따라서 이와 같은 한계를 극복하기 위해 Cancelable Biometrics라는 개념이 소개됨



01 연구 배경 및 목적

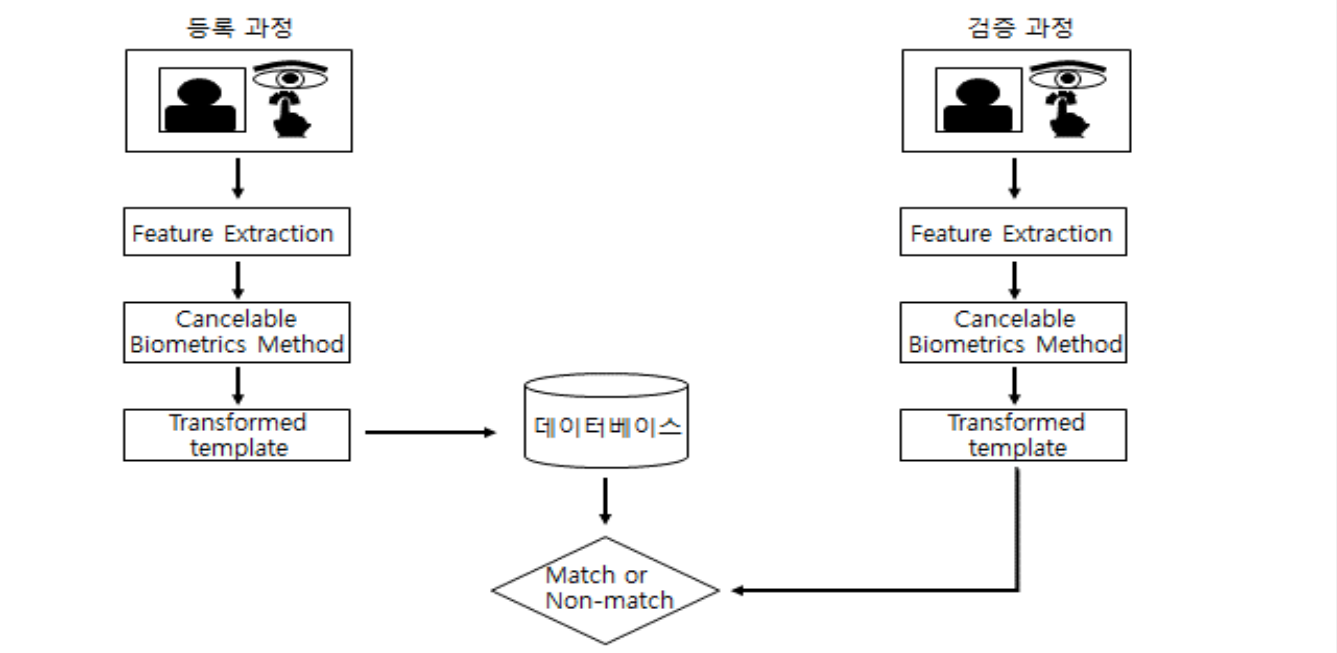
- Cancelable Biometrics란 특징 정보에 사용자별 보조 데이터를 혼합하거나 비가역 변환을 적용하여 생성된 변환된 형태의 템플릿을 통해 생체 기반 인증을 수행하는 방법

- 성능 보존(Preserving performance) : Cancelable Biometrics Method 적용 후의 정확도 성능은 적용 전의 정확도 성능을 보존해야 한다.
- 비가역성(Irreversibility) : Transformed template에서 원래의 특징 정보를 복원하는 것은 계산적으로 불가능해야 한다.
- 폐기가능성(Revocability) : Transformed template은 언제든지 폐기 또는 재생산이 가능해야 한다.
- 연결해제성(Unlinkability) : 같은 특징 정보로부터 생성된 Transformed template들 간에 상관관계는 없어야 한다.

01 연구 배경 및 목적

- 데이터베이스에 있는 template이 유출되어도 생체 데이터 보호 가능

- 패스워드 재설정 및 토큰 재발급과 같은 작업이 생체 기반 인증에서도 가능



01 연구 배경 및 목적

- 최근 Cancelable Biometrics를 위한 여러 방법들이 제안되고 있고 이에 따라 제안된 방법들이 여러 상황 및 보안 공격에 안전한지에 대한 연구 또한 필요

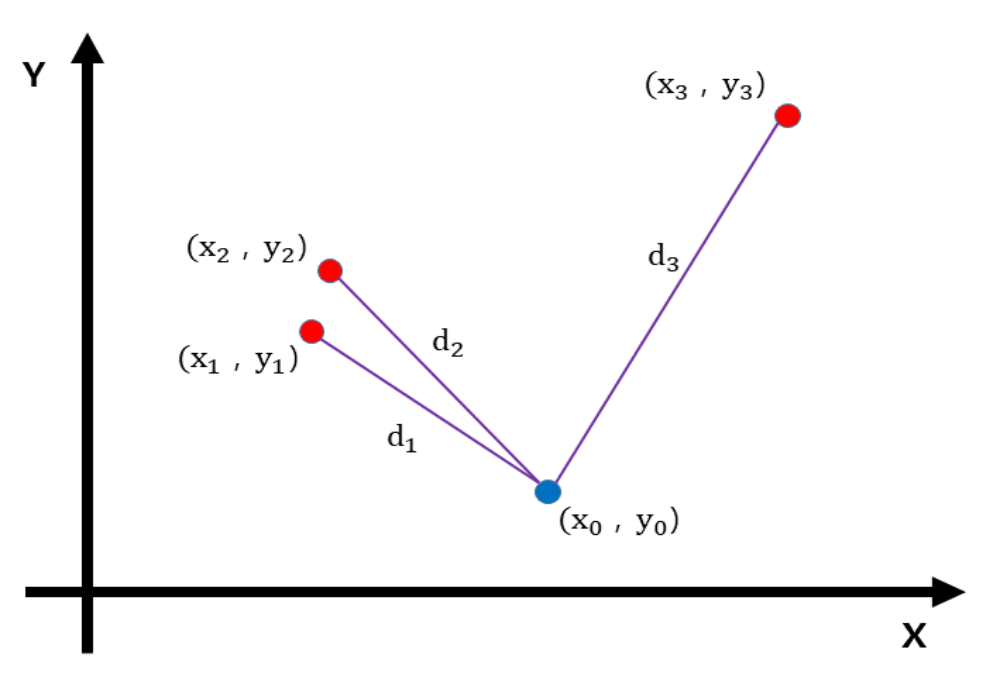
- 2019년 IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY에 게재된 Random Distance Method for Generating Unimodal and Multimodal Cancelable Biometric Features

- 본 연구의 목적은 Stolen helper data scenario에서 Random Distance Method에 대한 제 2 역상 공격을 제안하여 해당 Method가 다양한 상황에서도 안전한 생체 인증을 제공할 수 있는지 파악

02 Random Distance Method

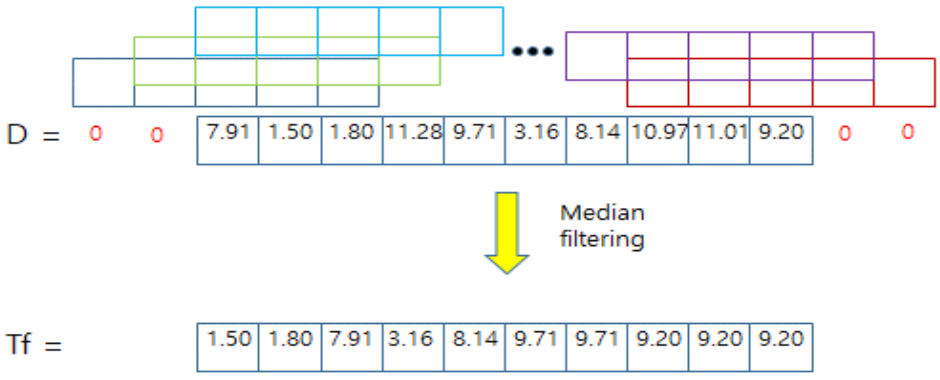
02 Random Distance Method의 개념 및 원리

- 특징 벡터 fv를 좌표계의 점인 FP로 표현
- 사용자별로 할당되는 Random Point와 FP의 거리를 계산
- 계산된 거리 값을 기반으로 여러 fv 간의 유사 여부 또는 일치 여부 판단

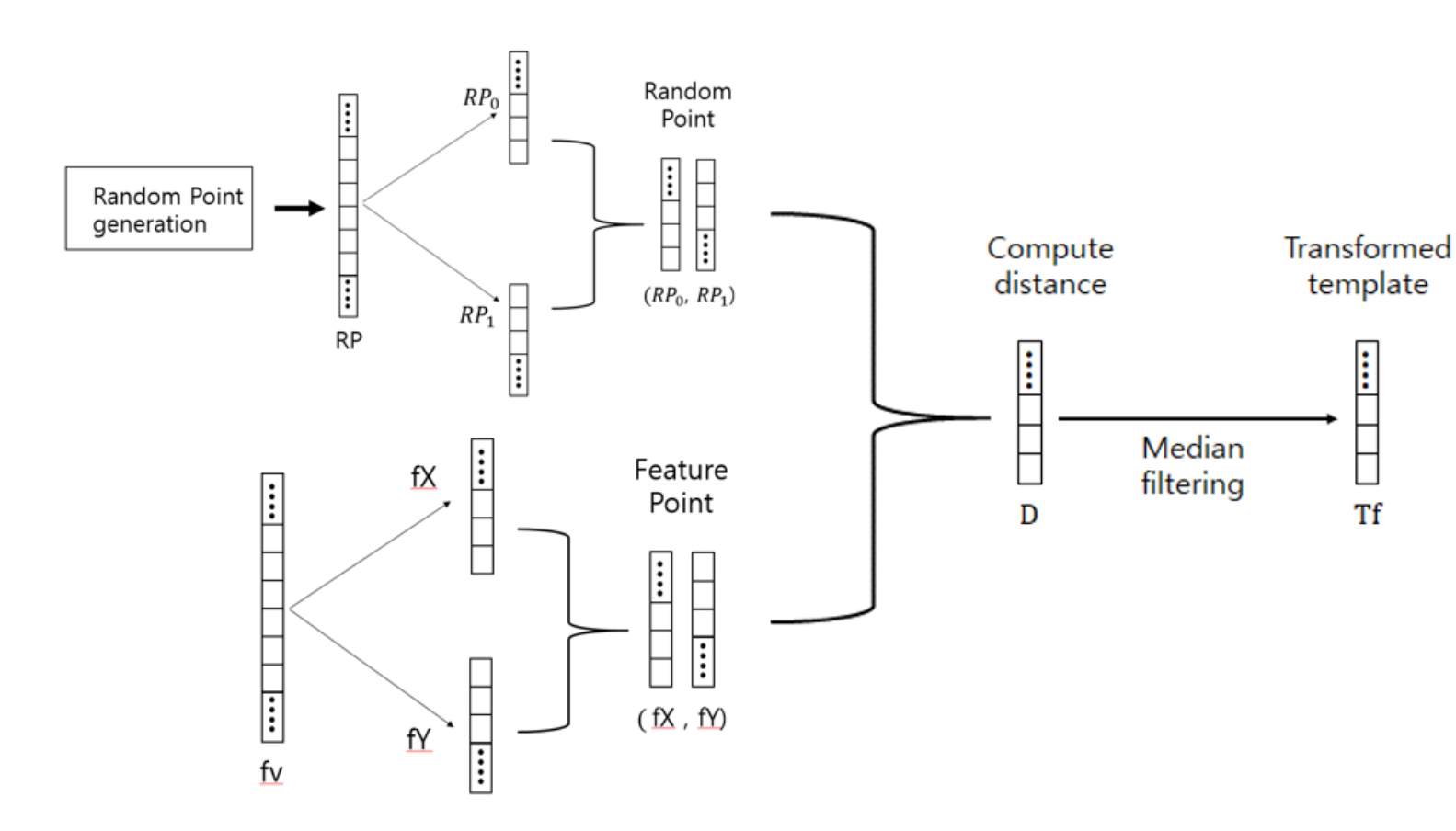


02 Random Distance Method의 개념 및 원리

- Median filter는 window size 범위 안의 값들을 크기 순으로 정렬한 후 중간 크기의 값을 출력하는 filter
- D벡터의 값들이 섞임과 동시에 일부 값들은 사라지고, 자라진 자리는 남은 값들이 채움
- Median filter를 통해 비가역성을 제공함과 동시에 거리 값을 인증 승인 및 거부를 판단하는데 사용할 수 있음



02 Random Distance Method의 구현 과정



02 Random Distance Method에 대한 실험 및 평가

데이터셋	사용된 특징 추출 기법	특징 벡터의 크기
IITD-E(귀)	[17]에서 구현된 pretrained CNN model	fv^{512}
CASIA-V5(얼굴)	[18]의 FEHash를 통한 전처리 과정과 pretrained FaceNet model	fv^{512}
FVC2002-DB1(지문)	[19]의 전처리 과정과 Kernel Principal Component Analysis	fv^{299}
CNU(걸음걸이)	[16]의 프레임워크	fv^{289}

<각 데이터셋에 대한 특징 추출 기법 및 특징 벡터의 크기>

window size	EER	DI	RI
3	2.55±0.82	3.65±0.24	94.95±2.49
5	5.00±1.53	3.13±0.21	86.85±5.40
7	6.75±1.27	2.81±0.17	80.10±7.34

<지문 데이터셋에 대한 RDM의 Matching performance>

03 RDM에 대한 제 2 역상 공격

03 Random Distance Method에 대한 제 2 역상 공격(Preimage attack)

- Preimage attack은 hash 함수와 같은 비가역 함수에 대한 공격 방법으로서 2가지 공격으로 나뉨

- 제 1 역상 공격 : 함수 결과 값이 주어져 있을 때, 그 결과 값을 출력하는 원본 입력 값을 찾아내는 공격

$$y = h(x)$$

- 제 2 역상 공격 : 원본 입력 값과 동일한 출력 값을 만들어내는 또 다른 입력 값을 찾아내는 공격

$$y = h(x) = h(x')$$

03 Random Distance Method에 대한 제 2 역상 공격(공격 상황)

- Cancelable Biometrics 분야에서는 제안된 방법에 대한 보안 공격 가능성을 분석하기 위해 크게 2가지 공격 상황을 가정

- Stolen helper data scenario: Transformed template과 해당 template를 생성할 때 사용했던 알고리즘 및 모든 파라미터를 공격자가 알고 있다는 가정

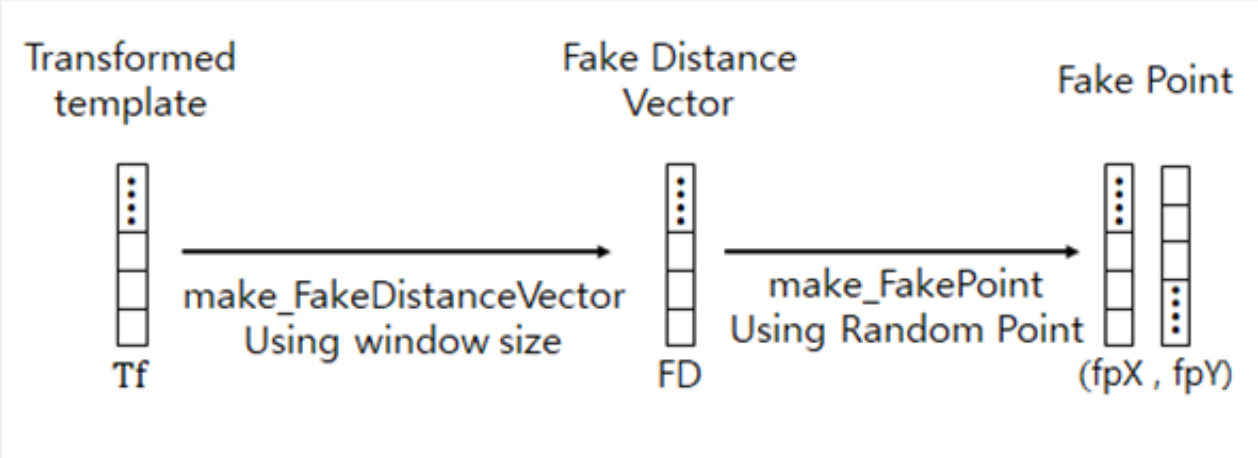
$$(Tf, Window\ size, RP)$$

- ARM(Attack via Record Multiplicity) : 원본 feature vector에 대한 여러 Transformed template과 해당 template들을 생성할 때 사용했던 파라미터 세트를 공격자가 원하는 만큼 수집할 수 있다는 가정

$$(Tf_1, Window\ size_1, RP_1), (Tf_2, Window\ size_2, RP_2), \cdots, (Tf_n, Window\ size_n, RP_n)$$

03 Random Distance Method에 대한 제 2 역상 공격(공격 설계)

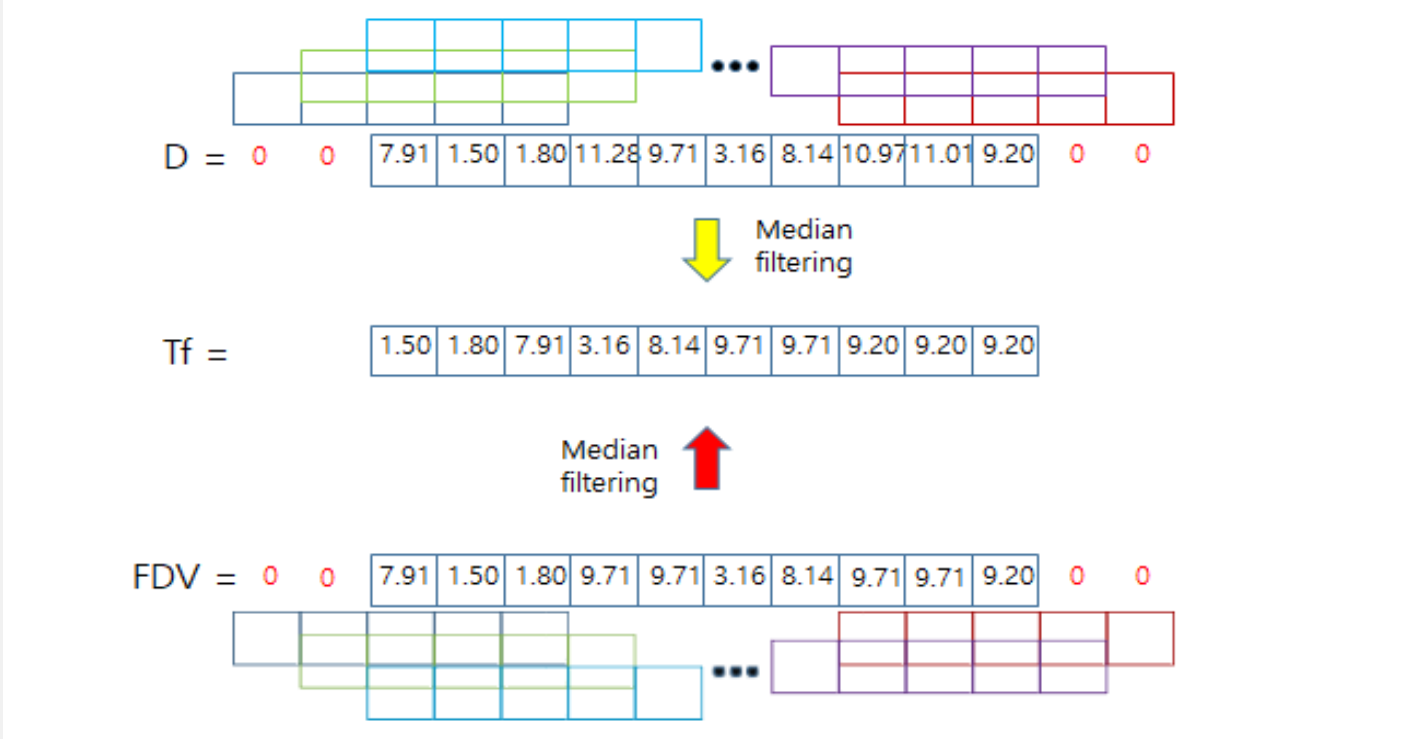
- 제안하는 공격은 make_FakeDistanceVector와 make_FakePoint 프로세스로 구성
- make_FakeDistanceVector는 median filter 적용 후의 결과가 Tf와 같은 Fake Distance Vector를 만드는 프로세스
- make_FakePoint는 Random Point와의 거리가 Fake Distance인 점을 찾아내는 프로세스



03 Make_FakeDistanceVector(아이디어)

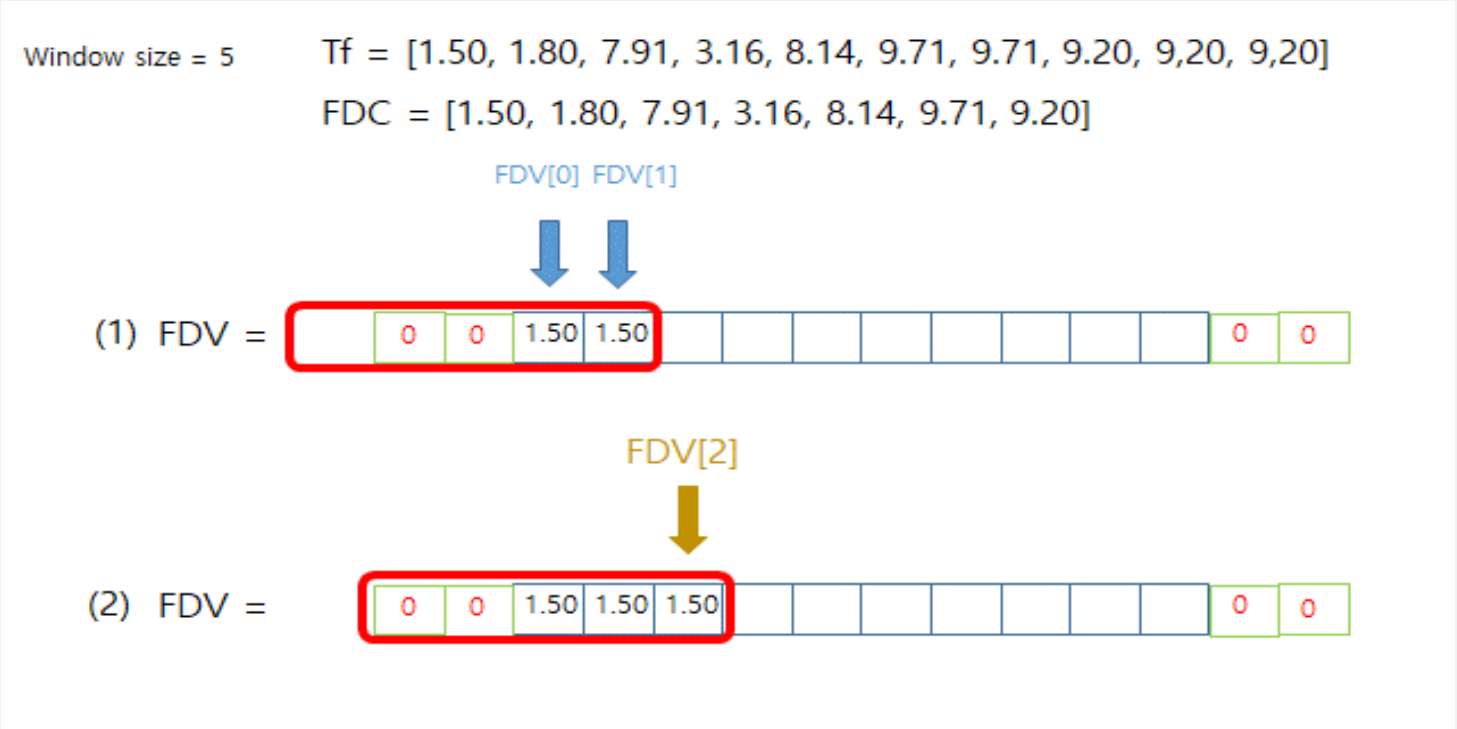
- make_FakeDistanceVector는 median filter 적용 후의 결과가 Tf와 같은 Fake Distance Vector를 만드는 프로세스(공격자는 Tf와 window size를 알고 있는 상황)

- Tf의 값들을 이용해서 각 index에 해당하는 window size 범위 내에 값들을 채움과 동시에 알맞게 ordering 시키는 것



03 Make_FakeDistanceVector(check_Promising)

- Tf 벡터의 값들에서 중복되는 값들을 제외하여 FDC(Fake Distance Candidate) 벡터 생성
- FDC의 값들을 자식 노드로서 백트래킹 기반 깊이 우선 탐색을 진행



03 Make_FakeDistanceVector(check_Promising)

- FDC의 값들을 자식 노드로서 백트래킹 기반 깊이 우선 탐색을 진행

Tf = [1.50, 1.80, 7.91, 3.16, 8.14, 9.71, 9.71, 9.20, 9,20, 9,20]

FDC = [1.50, 1.80, 7.91, 3.16, 8.14, 9.71, 9.20]

FDV = | 0 | 0 | 1.50 | 1.50 | 1.50 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Median_filtering(FDV)[0] == TF[0] <------- True or False ???

FDV = | 0 | 0 | 1.50 | 1.50 | 1.50 | 1.50 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Median_filtering(FDV)[1] == TF[1] <------- True or False ???

⋮

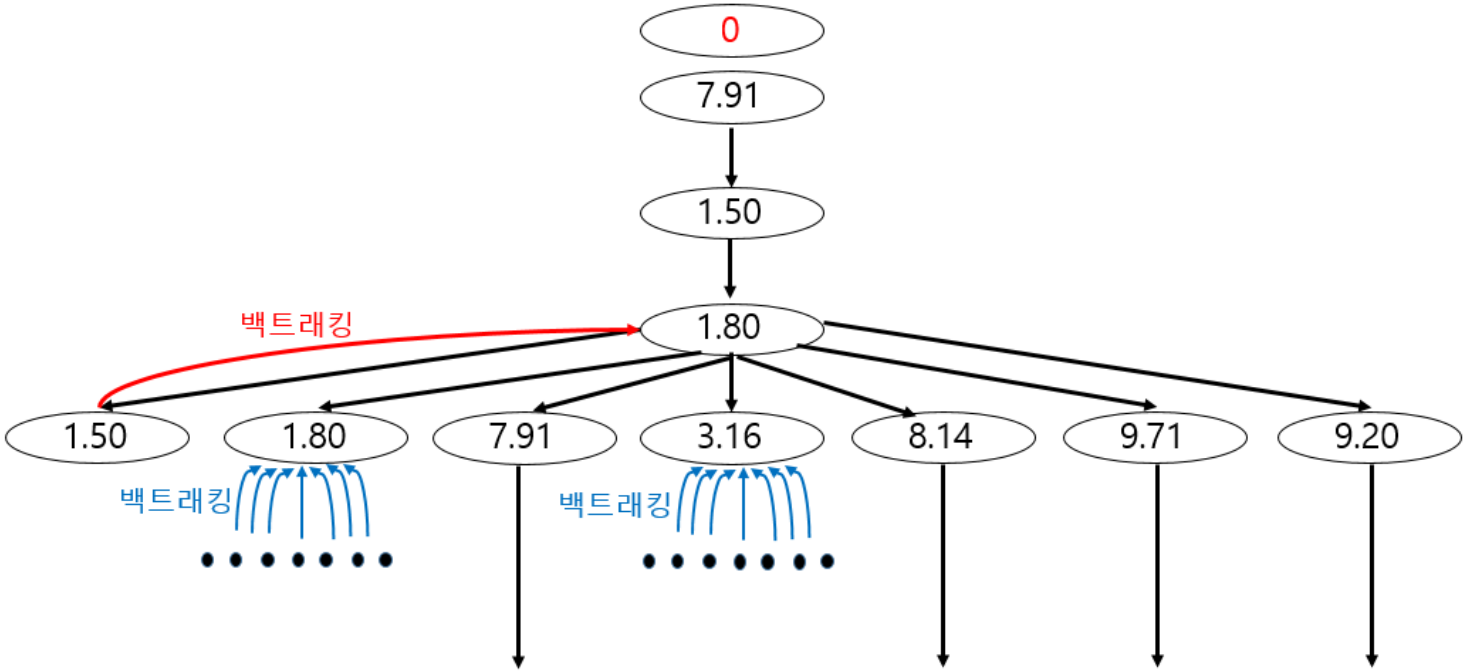
Median_filtering(FDV)[8] == TF[8] <------- True or False ???

Median_filtering(FDV) == TF <------- True or False ???

03 Make_FakeDistanceVector(check_Promising)

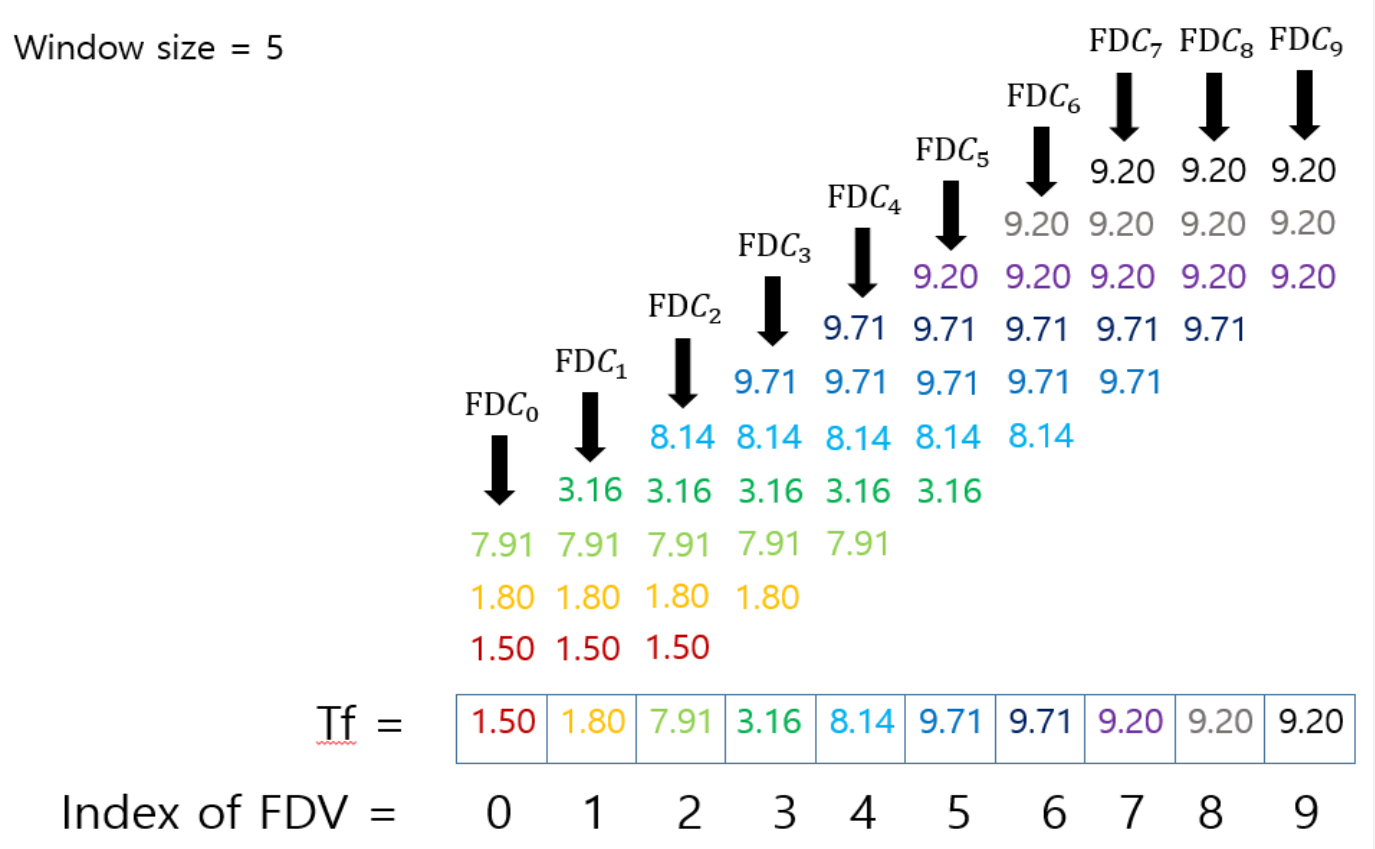
Tf = [1.50, 1.80, 7.91, 3.16, 8.14, 9.71, 9.71, 9.20, 9,20, 9,20]

FDC = [1.50, 1.80, 7.91, 3.16, 8.14, 9.71, 9.20]



03 Make_FakeDistanceVector(make_FDC)

- 탐색 공간을 줄이기 위해 FDV의 각 index별로 FDC를 만들어주는 것으로 최적화 작업을 진행



03 Make_FakeDistanceVector(make_FDC)

Tf = [1.50, 1.80, 7.91, 3.16, 8.14, 9.71, 9.71, 9.20, 9,20, 9,20]

FDC = [1.50, 1.80, 7.91, 3.16, 8.14, 9.71, 9.20]

FDV = | 0 | 0 | 1.50 | 1.50 | 1.50 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Median_filtering(FDV)[0] == TF[0] <------- True or False ???

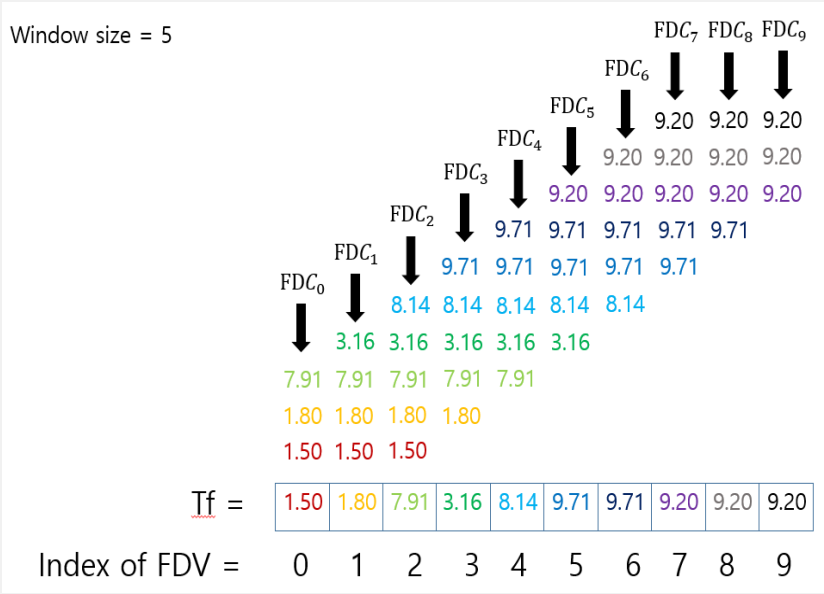
FDV = | 0 | 0 | 1.50 | 1.50 | 1.50 | 1.50 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Median_filtering(FDV)[1] == TF[1] <------- True or False ???

⋮

Median_filtering(FDV)[8] == TF[8] <------- True or False ???

Median_filtering(FDV) == TF <------- True or False ???



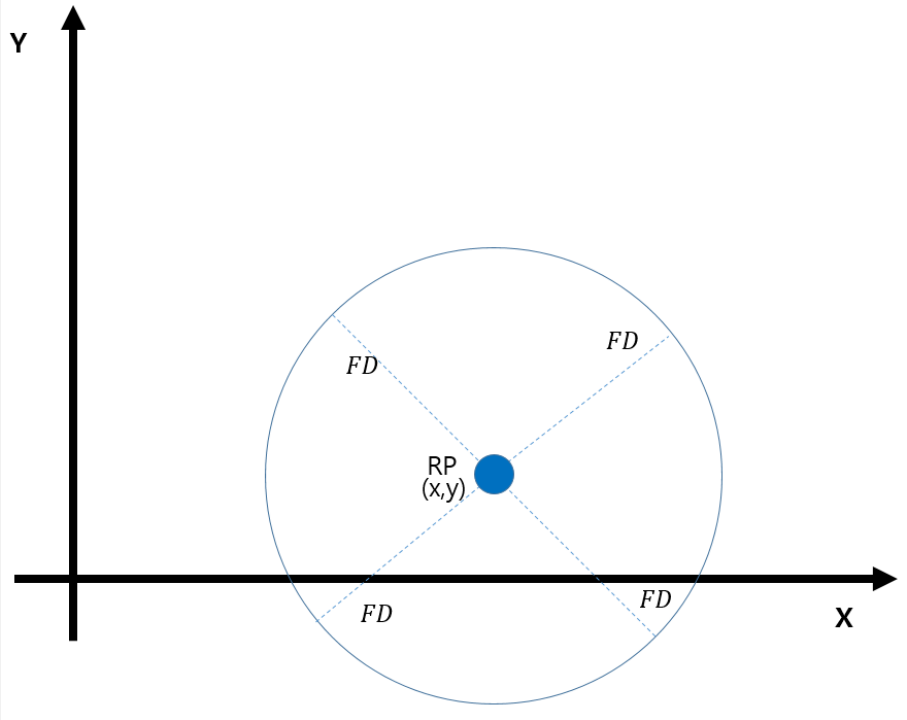
7^{10} = 282,475,249

3 * 4 * 5 * 5 * 4 * 4 * 3 * 2 * 2 * 1 = 57,600

03 Make_FakePoint(아이디어)

- make_FakePoint는 Random Point와의 거리가 Fake Distance인 점을 찾아내는 프로세스 (공격자는 FD와 Random Point를 알고 있는 상황)

- Random Point를 원의 중심으로 하고 반지름이 Fake Distance인 원을 그리게 되면 해당 원 위의 모든 점들은 Fake Point로서 쓰일 수 있다는 것



03 Random Distance Method에 대한 제 2 역상 공격(의사 알고리즘)

Algorithm 1: Check Promising (checkPromising)

Input : Transformed Template Tf, Fake Distnace Vector FDV, Index number Id, Window size w
Ouput: Decision

```
Temp = median_filter(FDV, w) ;
if (Id < (w - 1) / 2)
    return True;
else if (Id < len(Tf) - 1)
    if (Temp[Id - (w - 1) / 2] == Tf[Id - (w - 1) / 2])
        return True;
    else
        return False;
else
    if (Temp == Tf)
        return True;
    else
        return False;
```

Algorithm 2: Make Fake Distance Candidate for each index (makeFDC)

Input : Transformed Template Tf, Index number Id, Window size w
Ouput: Fake Distance Candidate FDC

```
FDC = [ ];
if (Id is the left edge)
    for i in range (Id + (w - 1) / 2)
        if (Tf[i] not in FDC)
            FDC.append(Tf[i]);
    return FDC;
else if (Id is the right edge)
    for i in range (Id - (w - 1) / 2, len(Tf) - 1)
        if (Tf[i] not in FDC)
            FDC.append(Tf[i]);
    return FDC;
else
    for i in range (Id - (w - 1) / 2, Id + (w - 1) / 2)
        if (Tf[i] not in FDC)
            FDC.append(Tf[i]);
    return FDC;
```

03 Random Distance Method에 대한 제 2 역상 공격(의사 알고리즘)

Algorithm 3: Make Fake Distance Vector (makeFDV)

Input : Transformed Template Tf, Index number Id, Window size w
Ouput: Fake Distance Vector FDV
global stop flag;
stop flag = False;
FDV = [0] $*$ len(Tf);
if (len(Tf) == Id)
    stop flag = True;
    return FDV;
FDC = make_FDC(Tf, Id, w);
for $candidate$ in FDC
    FDV[Id] = $candidate$;
if (check_Promising(Tf, FDV, Id, w))
    make_FDV(Tf, Id + 1, w);
if (stop flag == True)
    break;

Algorithm 4: Make Fake Point (makeFP)

Input : Fake Distance Vector FDV, The x coordinates of Random Point RPx, The y coordinates of Random Point RPy
Ouput: The x coordinates of Fake Point FPx, The y coordinates of Fake Point FPy
FPx = [ ];
FPy = [ ];
for i in range (len(FDV) - 1)
    FPx.append(RPx[i]);
    FPy.append(RPy[i] + FDV[i]);
return FPx, FPy;

03 Random Distance Method에 대한 제 2 역상 공격(공격 수행 시간)

window size	IITD-E	CASIA-V5	2002_DB1	CNU
3	0.079±0.024	0.08±0.019	0.028±0.002	0.032±0.003
5	0.518±0.089	0.537±0.037	0.224±0.029	0.204±0.038
7	7.928±2.179	8.1±2.54	2.343±0.483	1.898±0.763

각 데이터셋의 Transformed template에 대한 공격 수행 시간(sec)

04 결론 및 향후 계획

04 결론 및 향후 계획

결론

- 본 연구를 통해 Stolen helper data scenario에서 Random Distance Method에 대한 제 2 역상 공격이 가능함을 보임

- 이와 같은 공격을 통해 공격자는 등록된 사용자의 인증 정보로 시스템에 접근이 가능

향후 연구

- ARM 상황에서의 Random Distance Method에 대한 공격 연구가 필요

- 본 논문에서 제시한 공격 및 해당 취약점을 보완하기 위한 Random Distance Method에 대한 연구 또한 필요

감사합니다