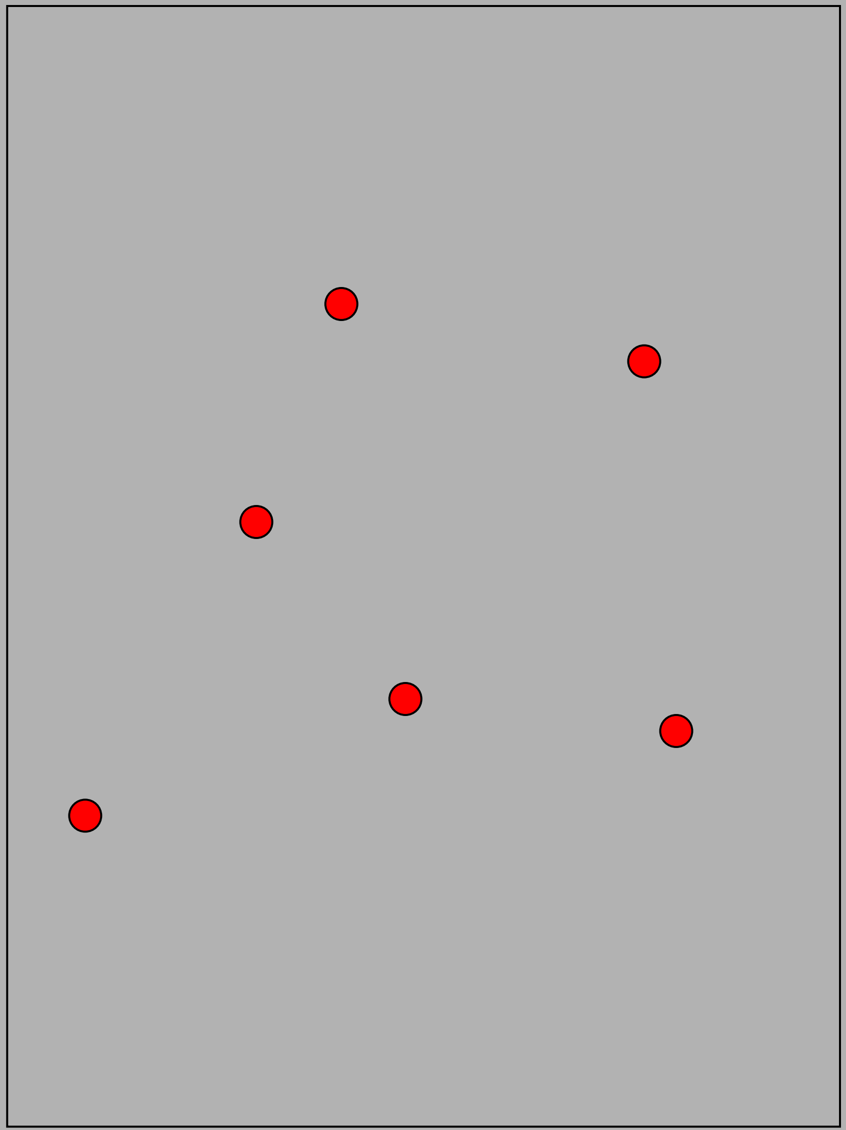
Finding the Original Point Set Hidden among Chaff

Ee-Chien Chang Ren Shen Francis Weijian Teo

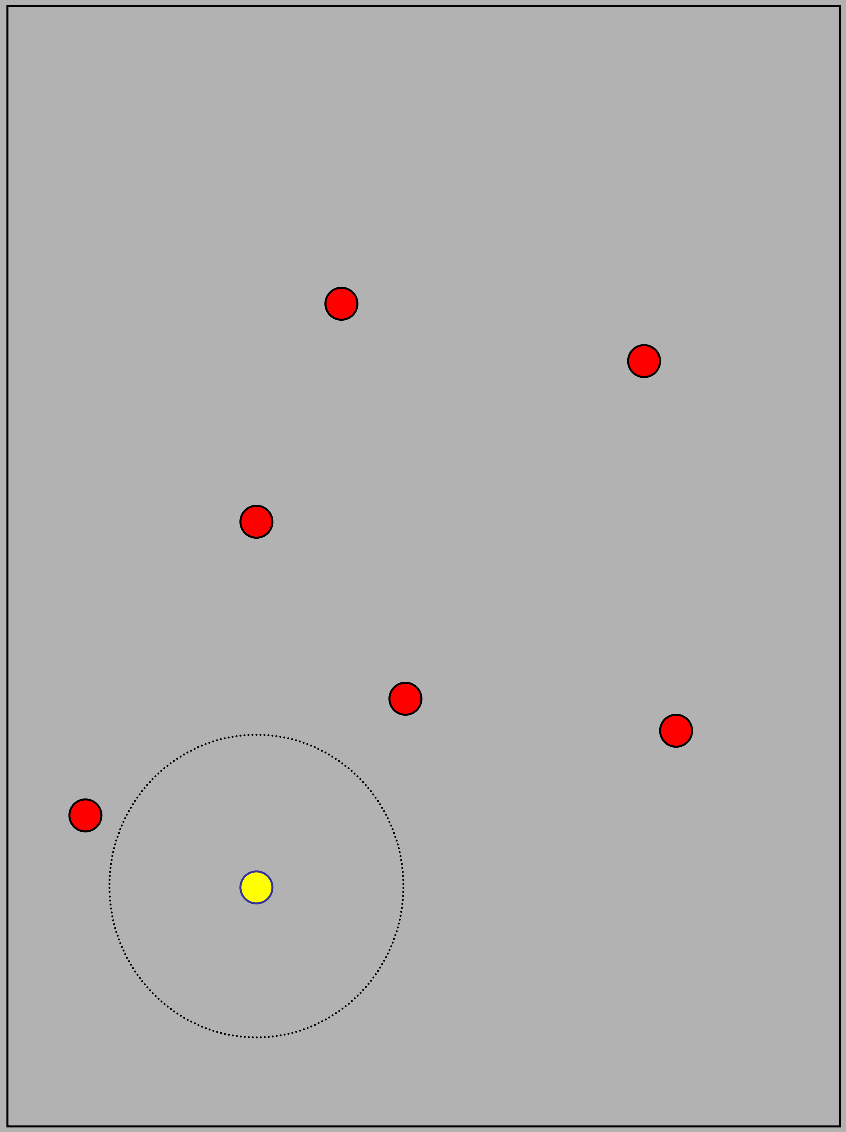
Department of Computer Science
National University of Singapore

The problem



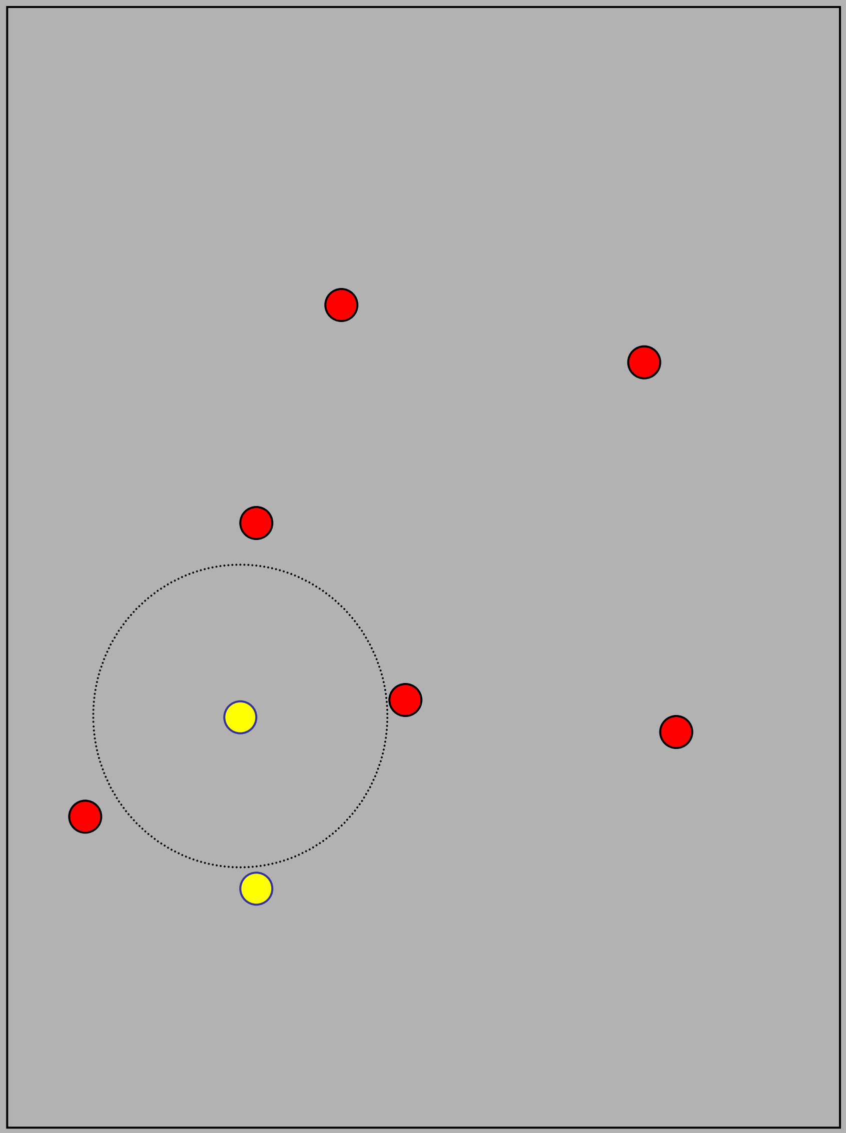
The original point set is a set of 2-d points in bounded domain.

The problem



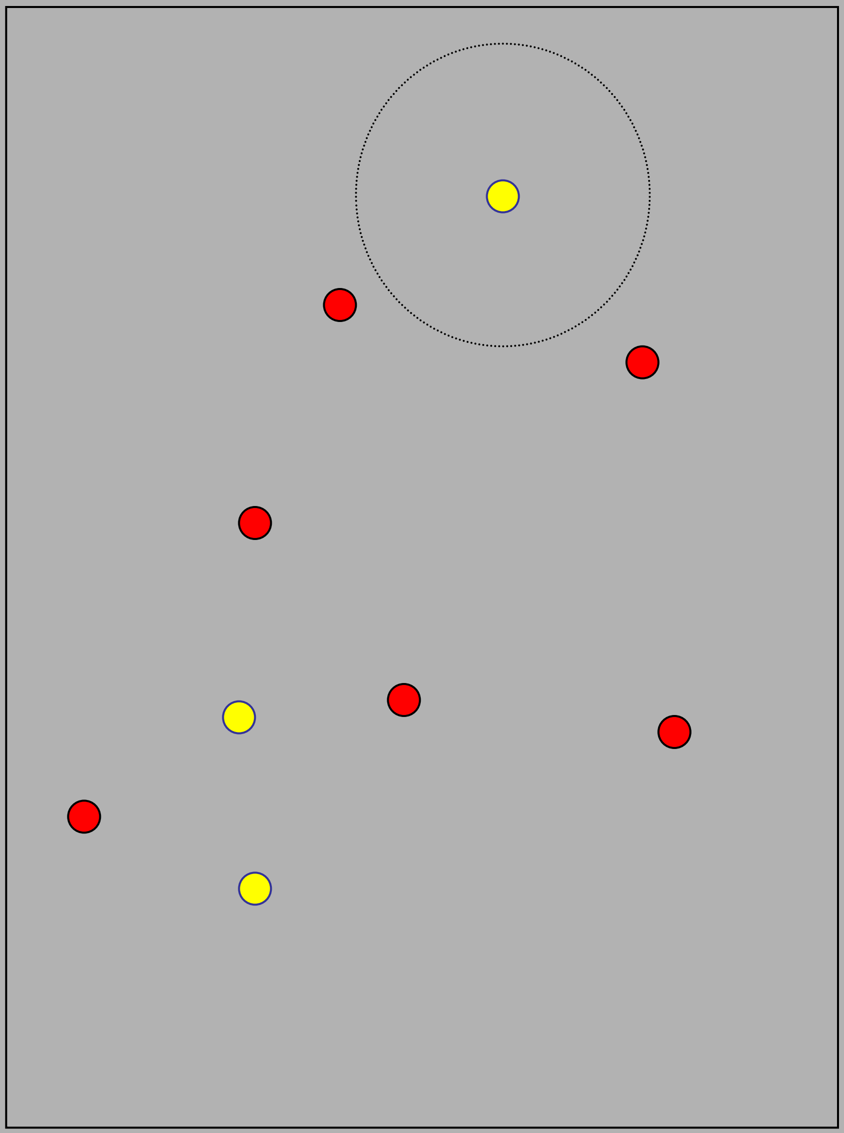
To hide the original,
add a chaff point that is
not close to any original.

The problem



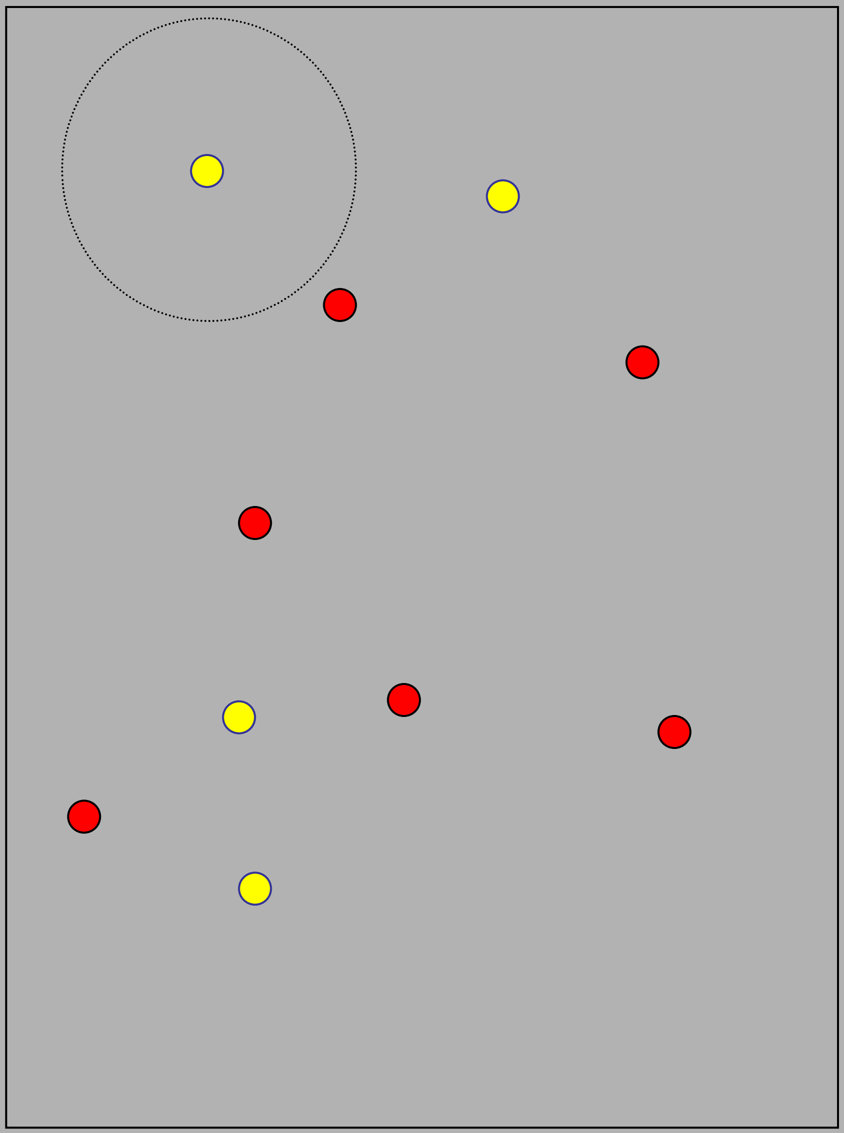
To hide the original,
add a chaff point that is
not close to any original
and previously added point.

The problem



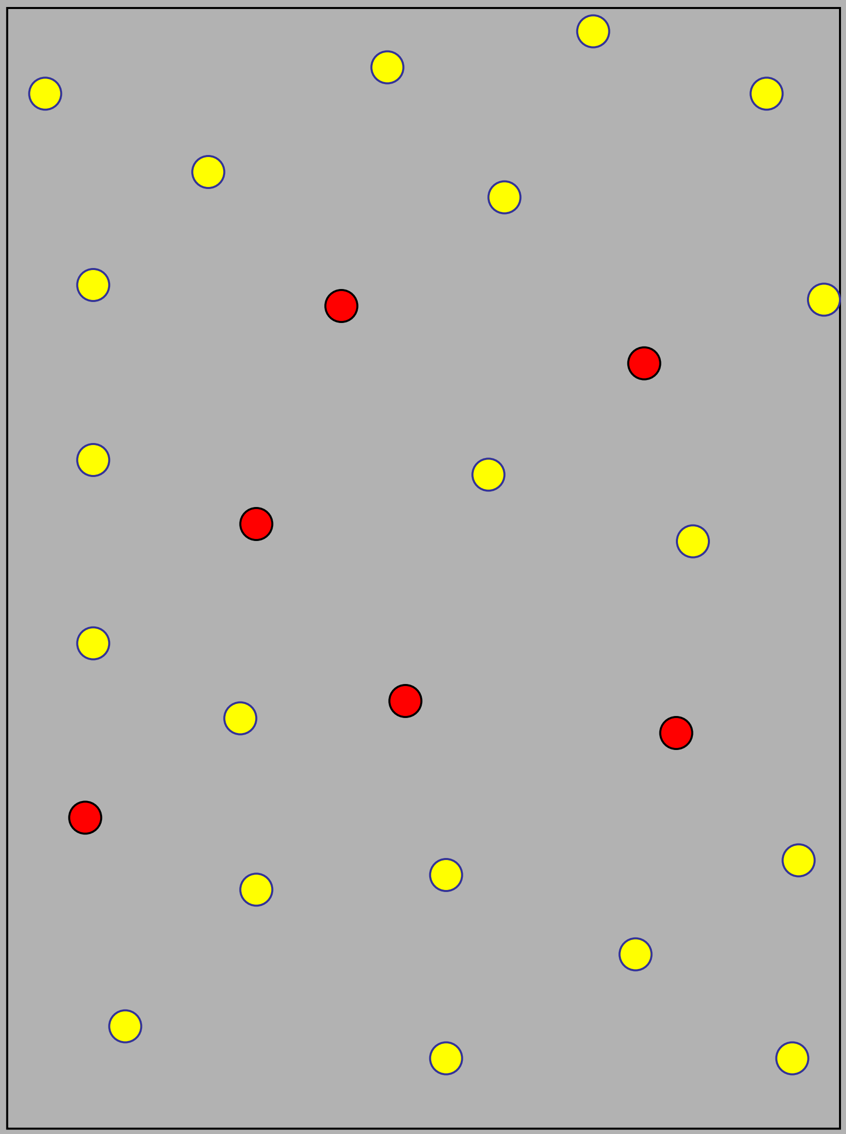
To hide the original,
add a chaff point that is
not close to any original
and previously added point.

The problem



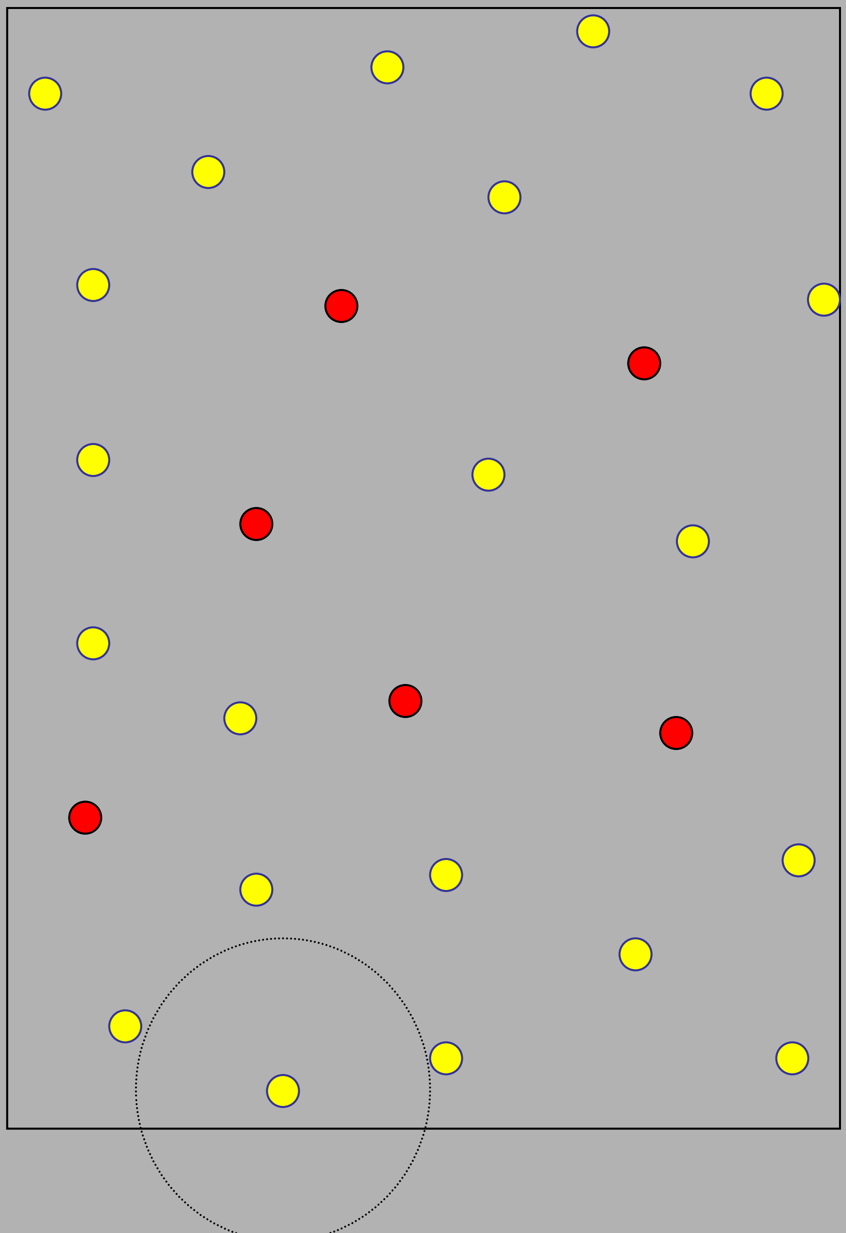
To hide the original,
add a chaff point that is
not close to any original
and previously added point.

The problem



To hide the original,
add a chaff point that is
not close to any original
and previously added point.

The problem

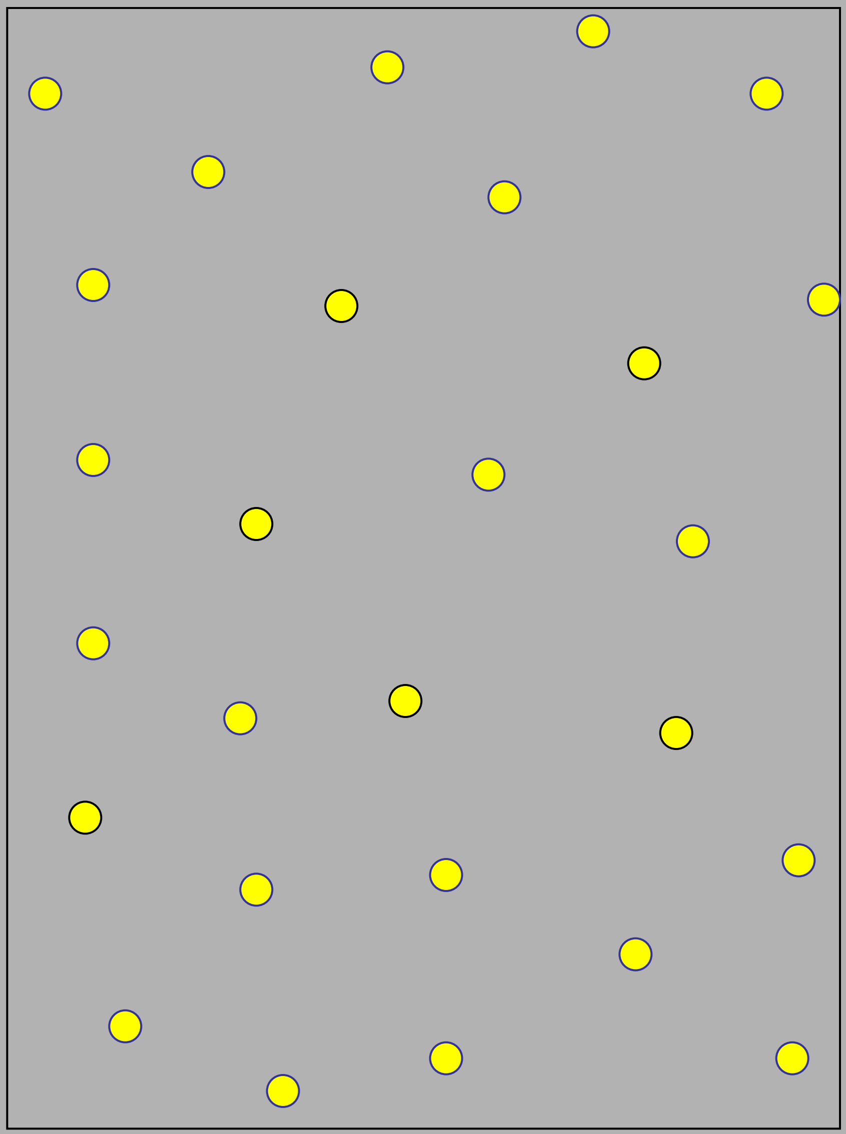


Keeps adding until
it is impossible to do so

The final point set is *well-separated.* That is, no 2 points are too close to each other.

Online parking.
(RNYI 1958)

The problem



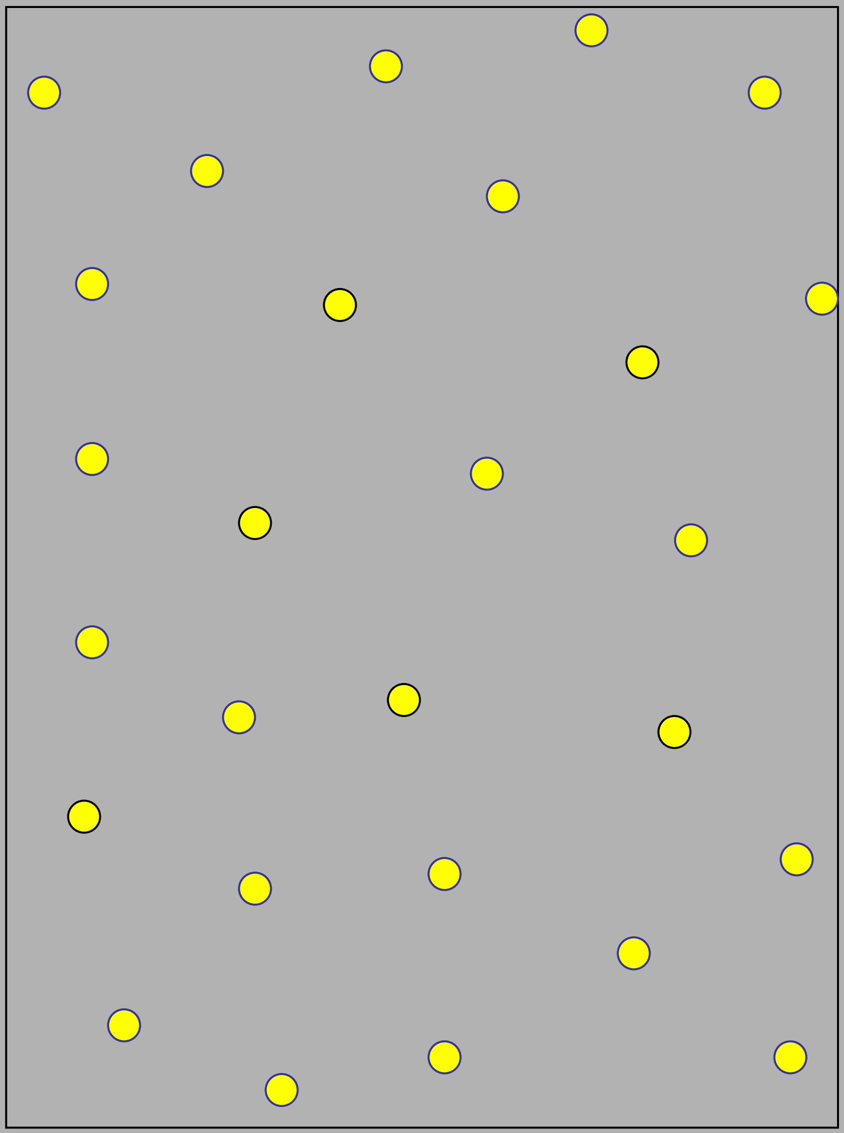
What the adversary get is...

The adversary wants to find out the original.

Since there are a total of **27** points, and the original contains **6** points,
if the adversary make a "random guess", the chances of success is

$$\binom{27}{6}^{-1}$$

The problem

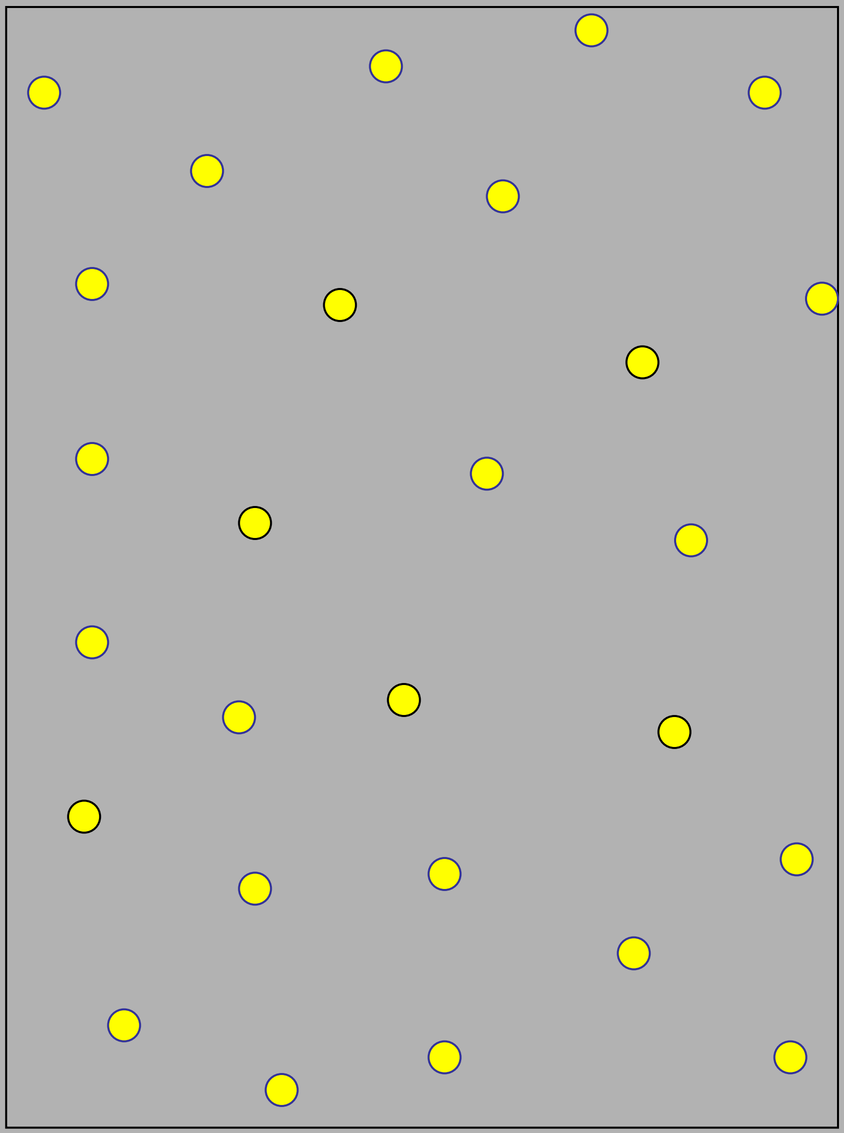


In this paper, we act as the adversary.

We want to guess the original better than

$$\binom{27}{6}^{-1}$$

The problem



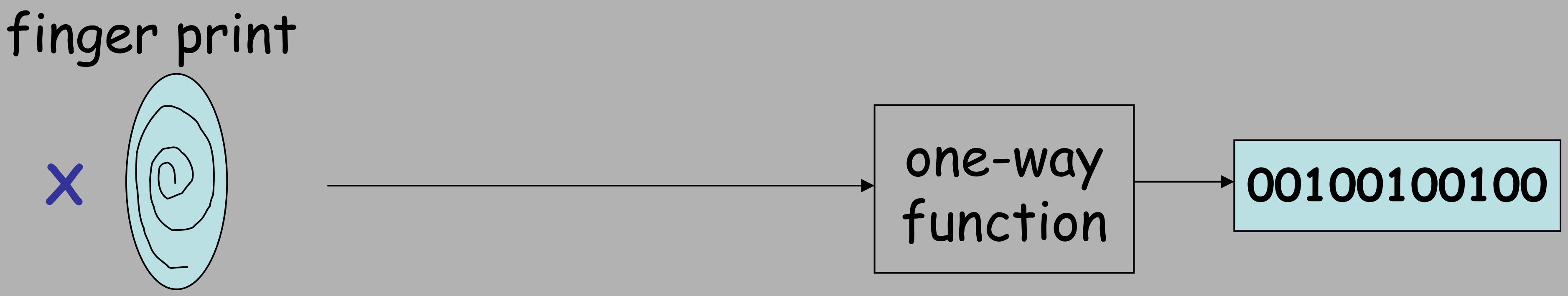
In this paper, we act as the adversary.

We want to guess the original better than

$$\binom{27}{6}^{-1}$$

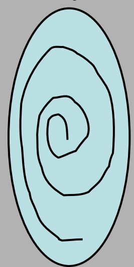
Motivation: *Secure Sketch for Fingerprint*

Background: Secure Sketch



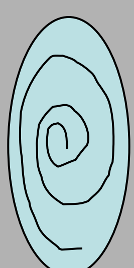
Background: Secure Sketch

finger print

X  → one-way function → 00100100100

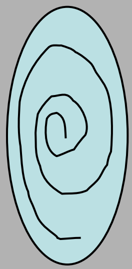
$0 < d(X, Y) < \varepsilon$

Another scan of the same finger

Y 

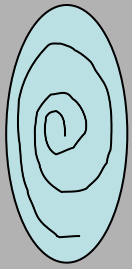
Background: Secure Sketch

finger print

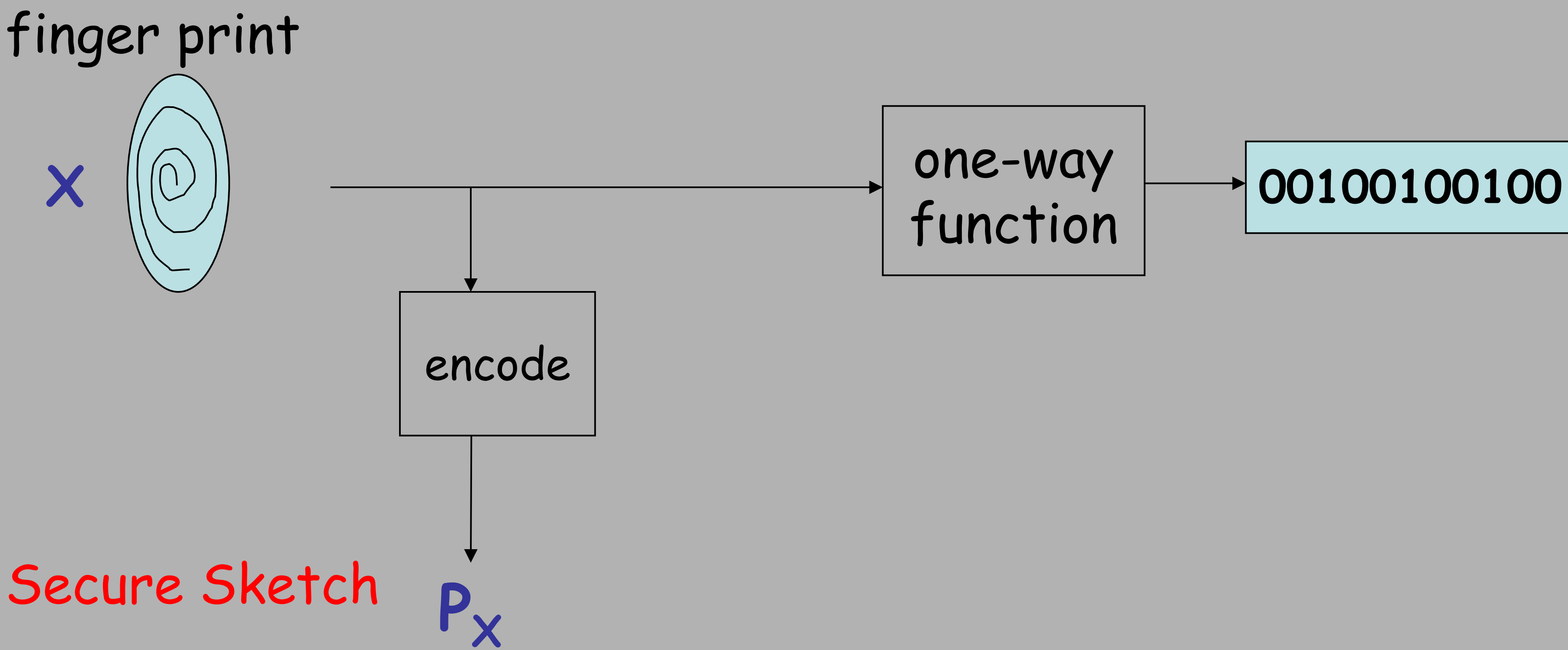
X  → one-way function → 00100100100

$0 < d(X, Y) < \varepsilon$ $\qquad\qquad$ $\neq$

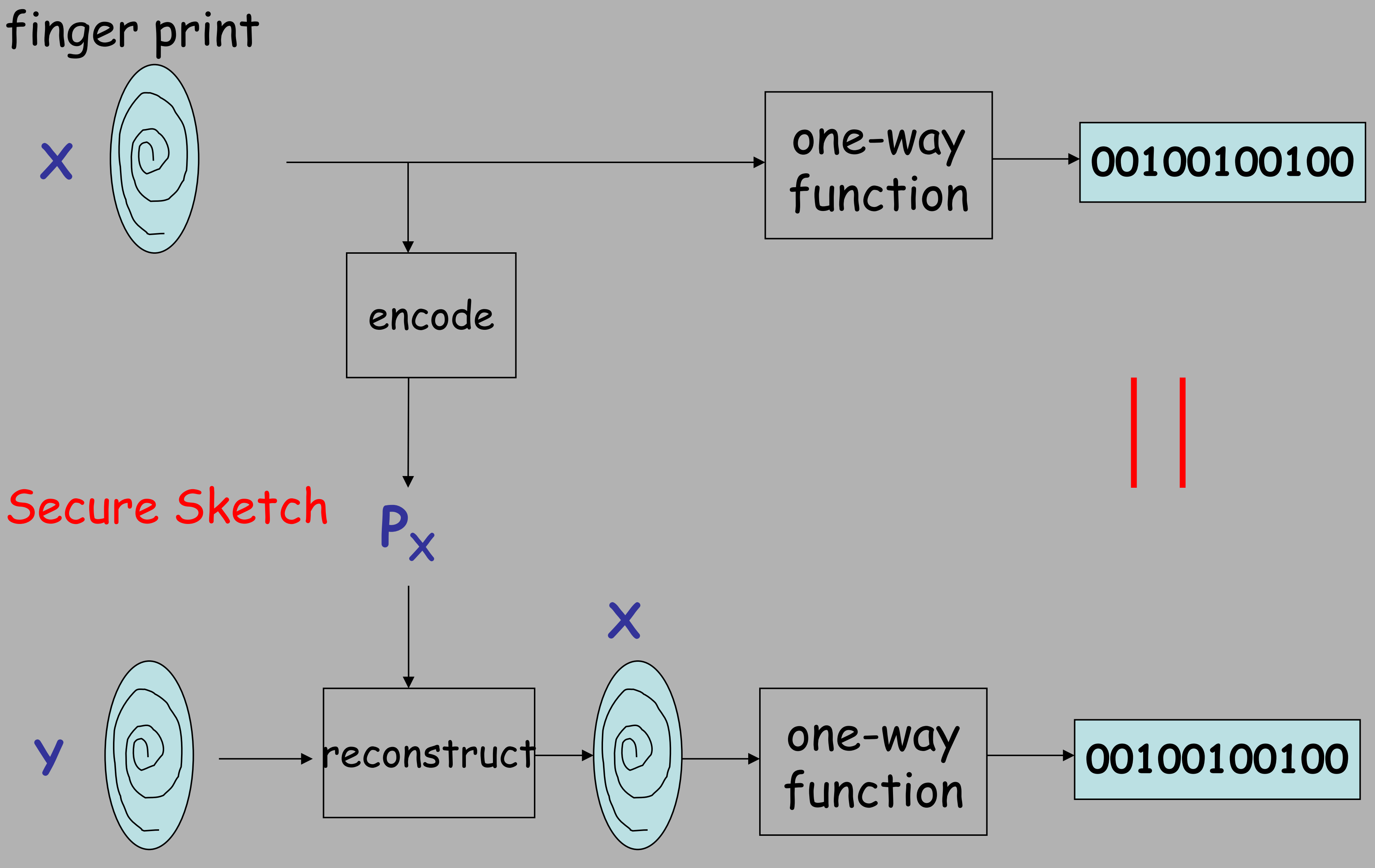
Another scan of the same finger

Y  → one-way function → 11100110011

Background: Secure Sketch P_x

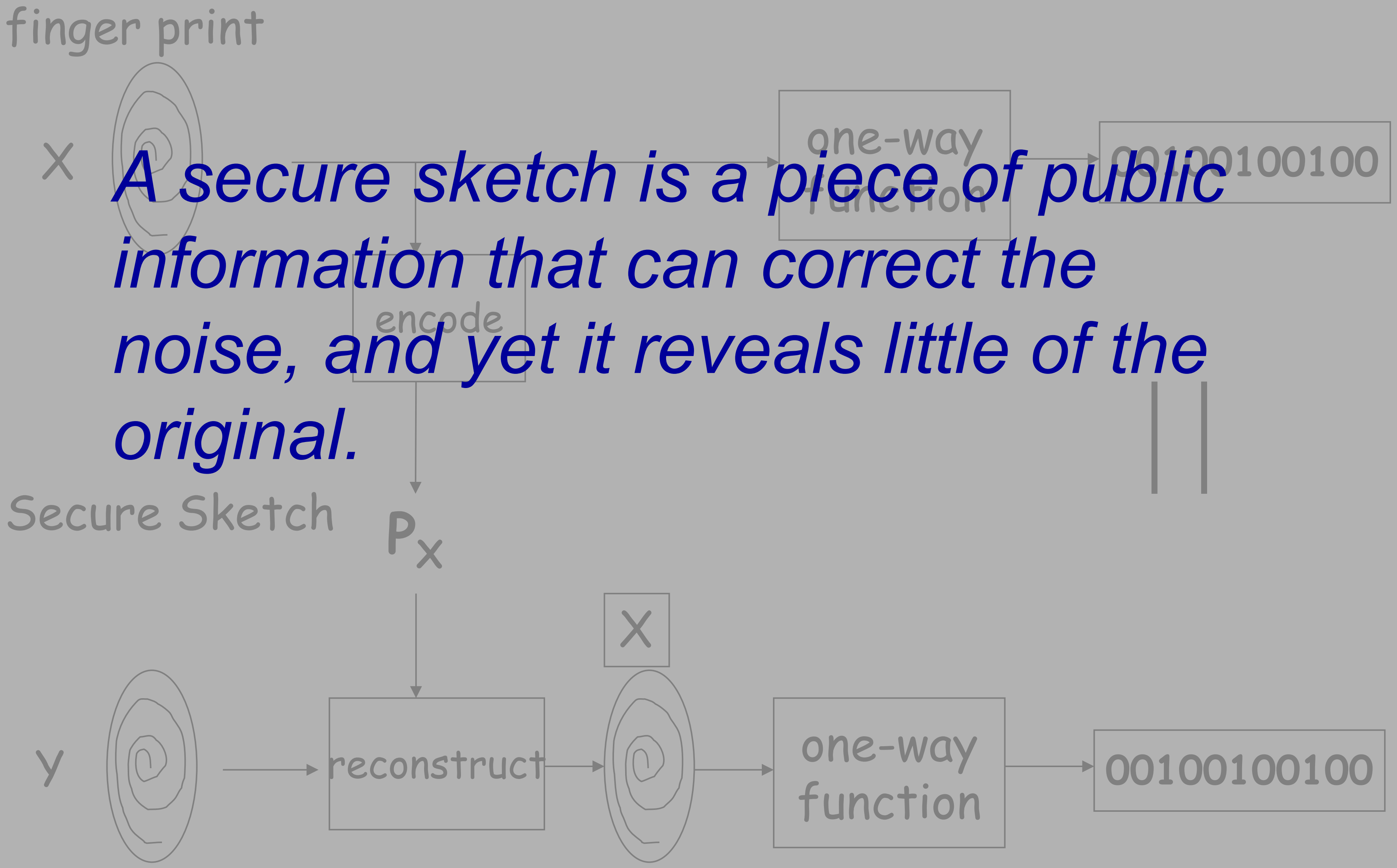


Background: Secure Sketch P_x

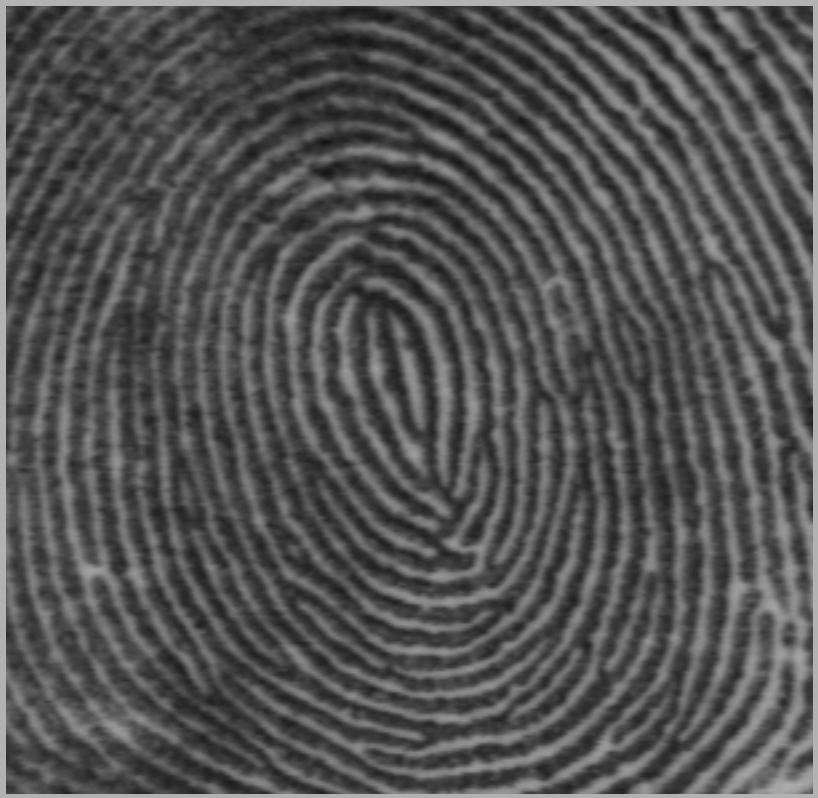


Background: Secure Sketch P_x

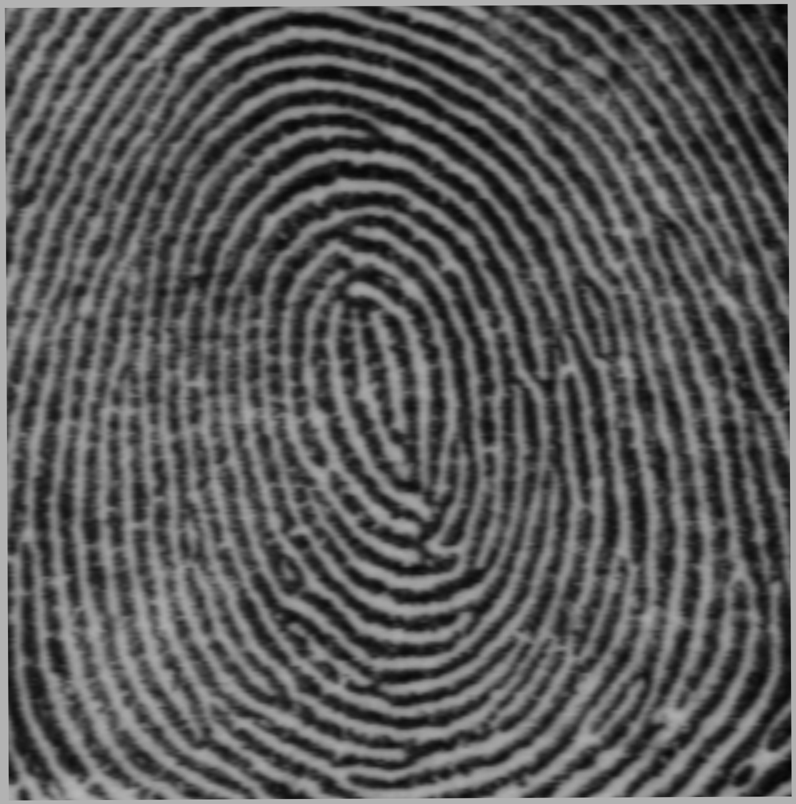
A secure sketch is a piece of public information that can correct the noise, and yet it reveals little of the original.



Background: Fingerprint

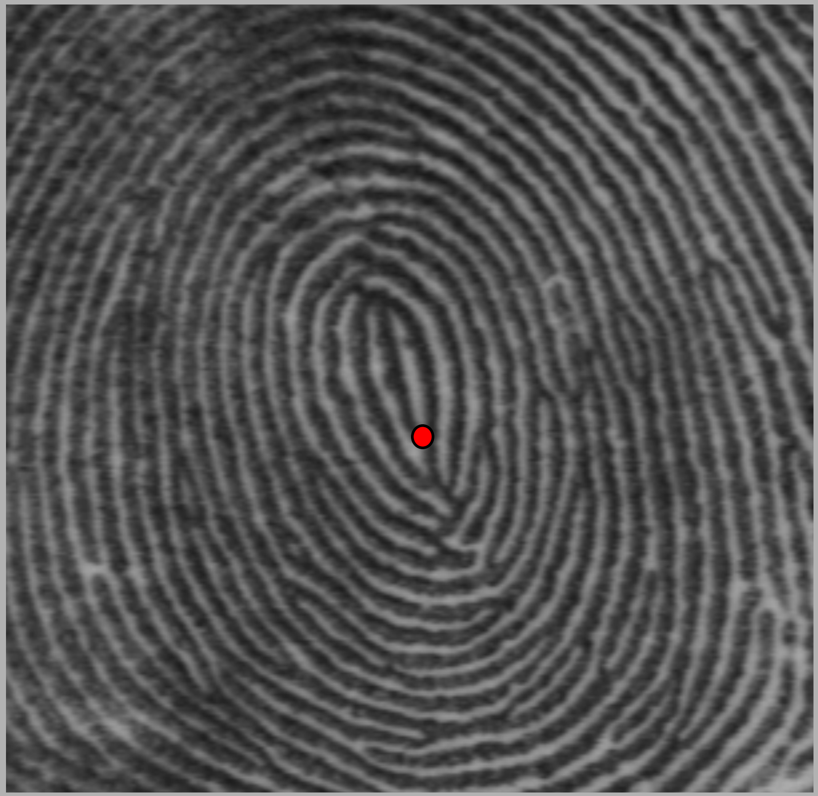


First scan of a finger

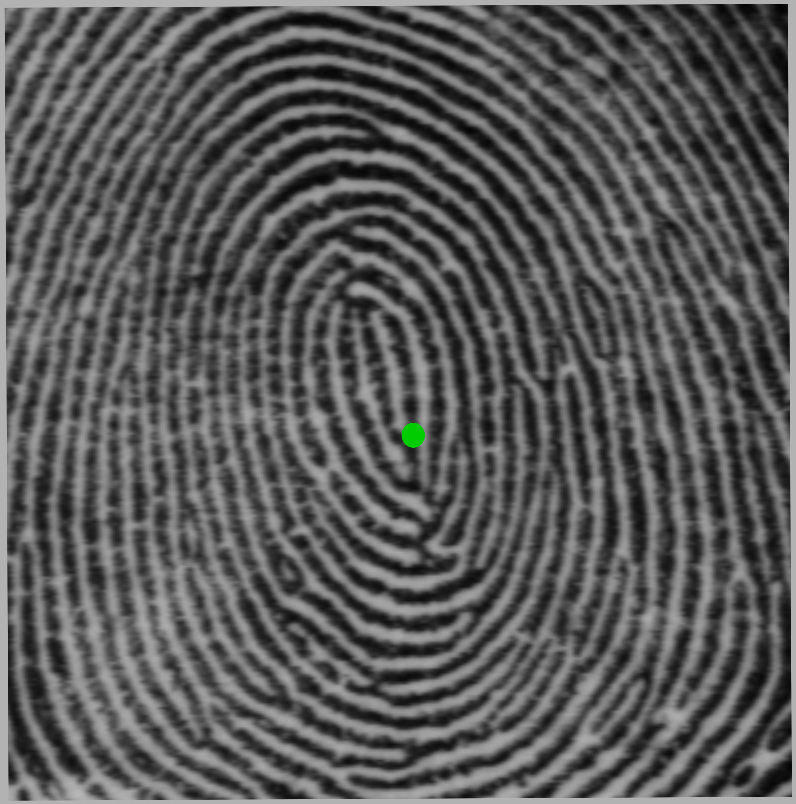


Another scan of the same finger

Background: Fingerprint

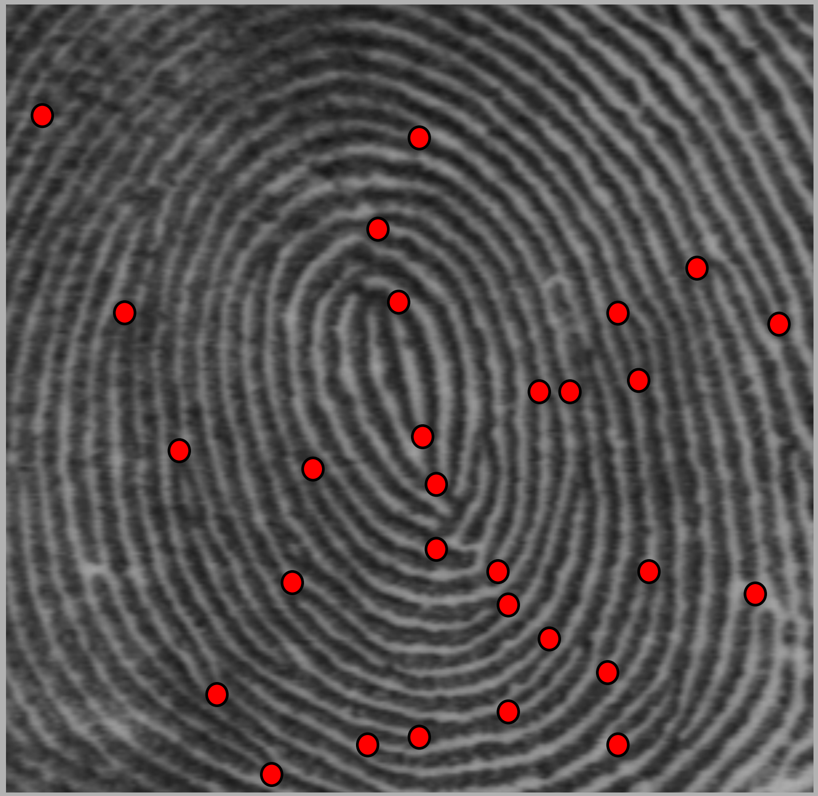


Point set X

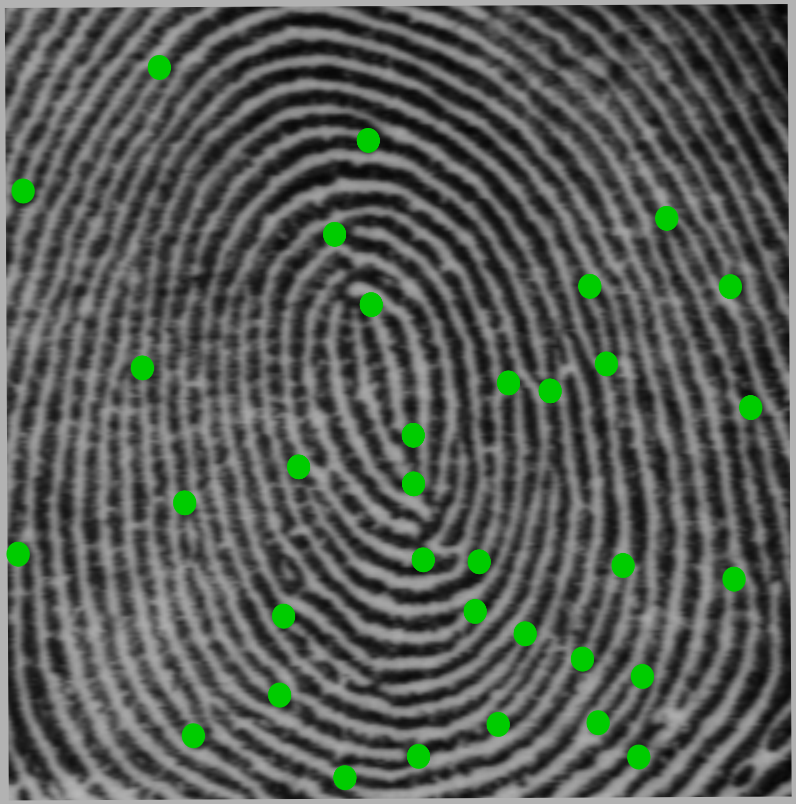


Point set Y

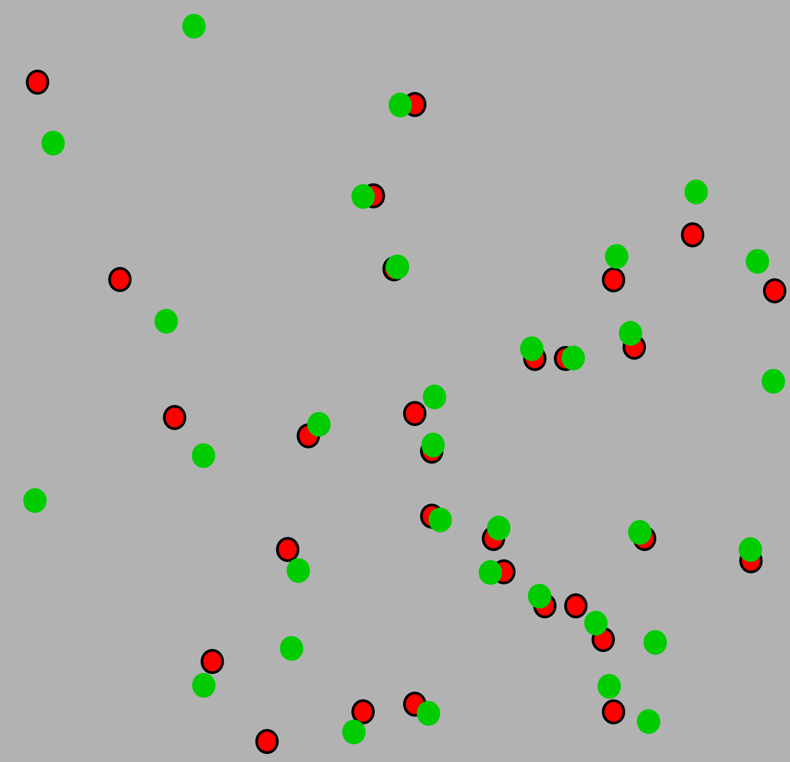
Background: Fingerprint

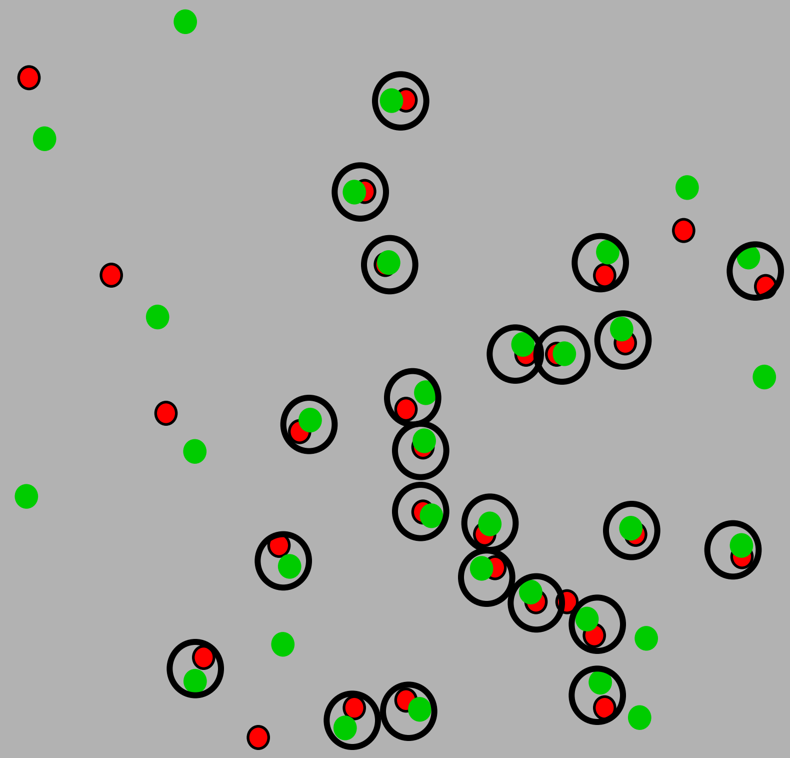


Point set X



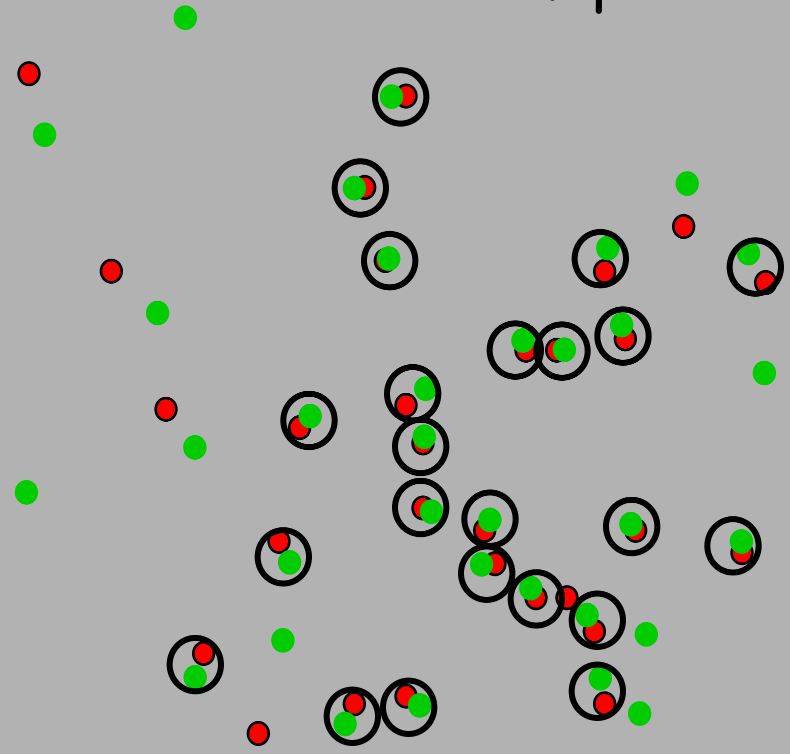
Point set Y





X and Y are similar if Y can be obtained from X by

1) perturb each point in X by a small amount.
2) replace at most t number of points.

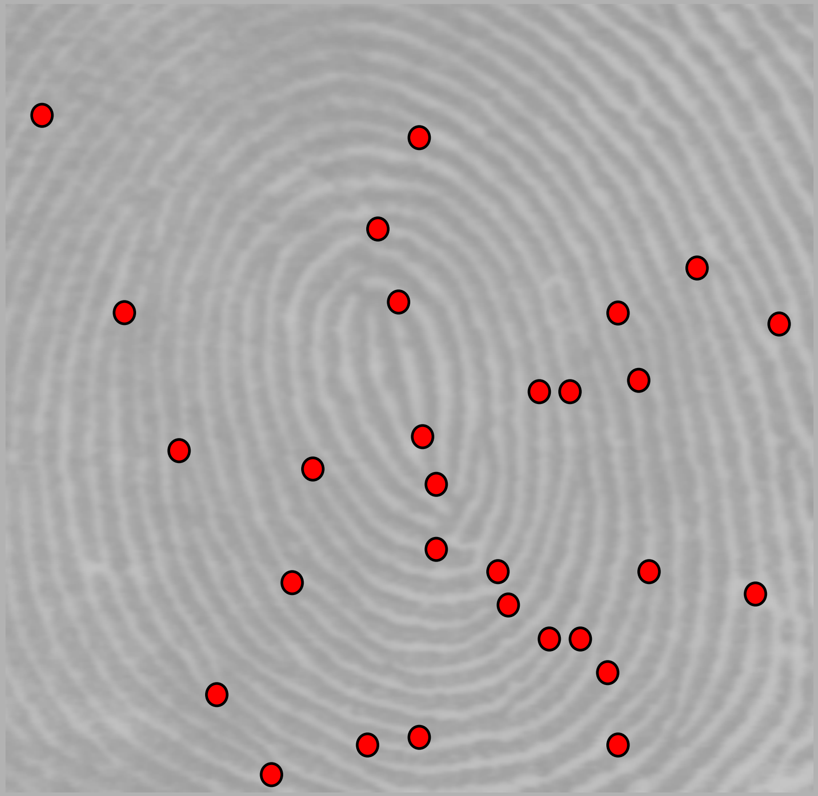


Now, we want a secure sketch that can correct the above 2 types of noise.

Background: Chaff (Clancy et. al.)

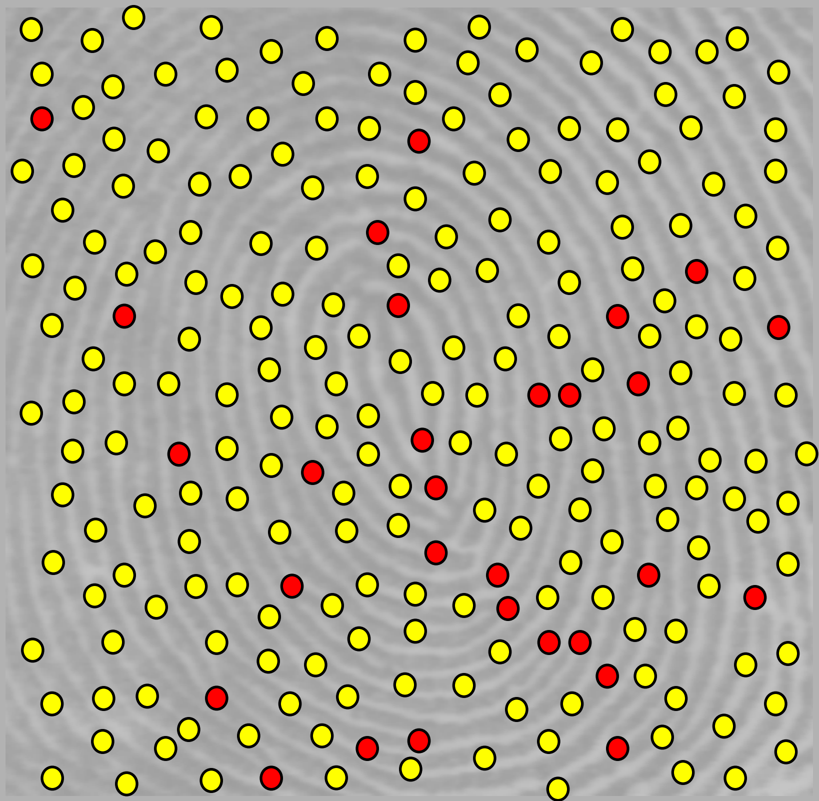
Secure sketch consists of two parts.

Part 1: Given the original point set **X**.



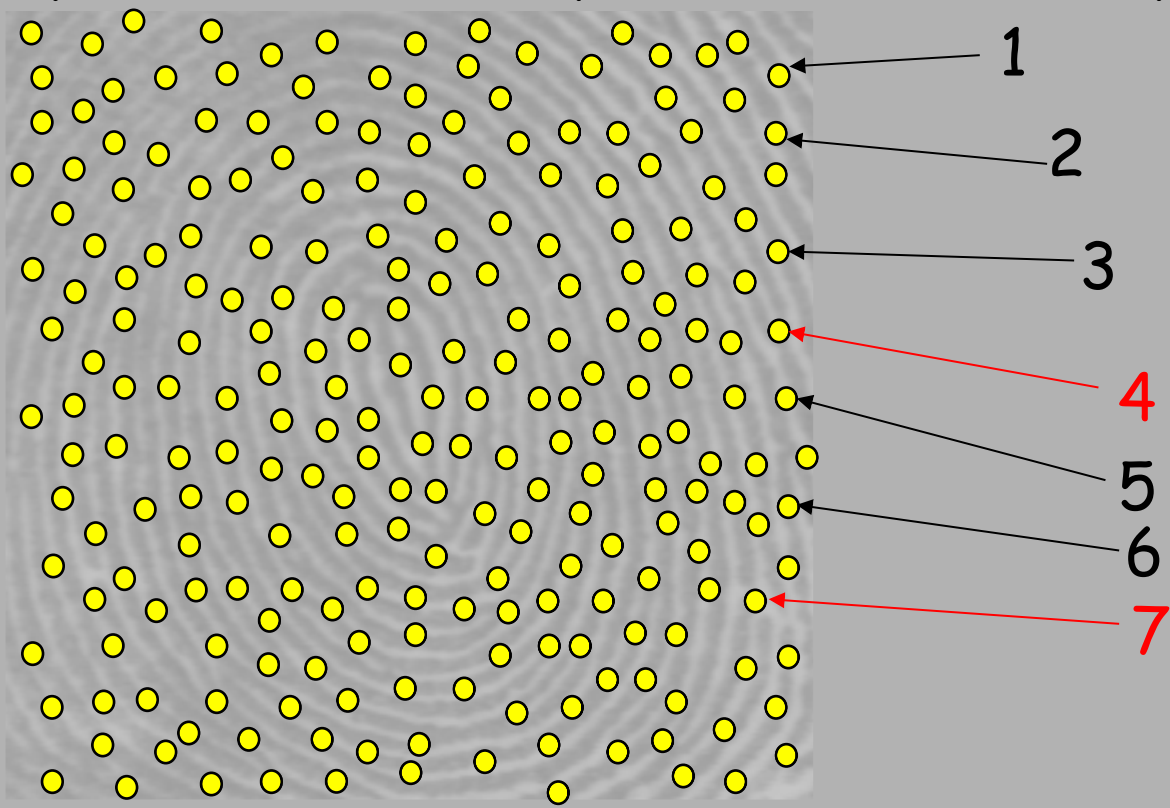
Background: Chaff (Clancy et. al.)

Part 1: Given the original point set X, using online parking to generate more points.



Background: Chaff (Clancy et. al.)

Part 1: Given the original point set X, using online parking to generate more points. The final point set is the 1st part of the sketch.



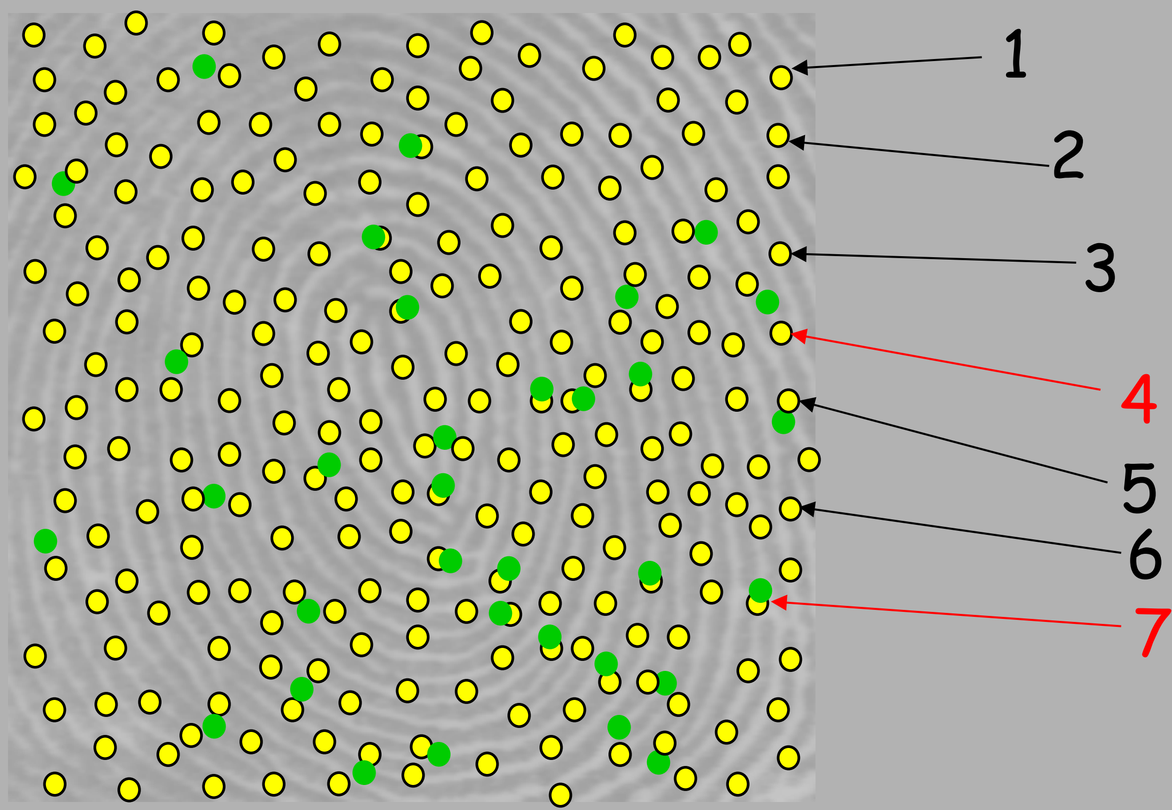
R

Indices of X are : 4, 7, …

Background: Chaff (Clancy et. al.)

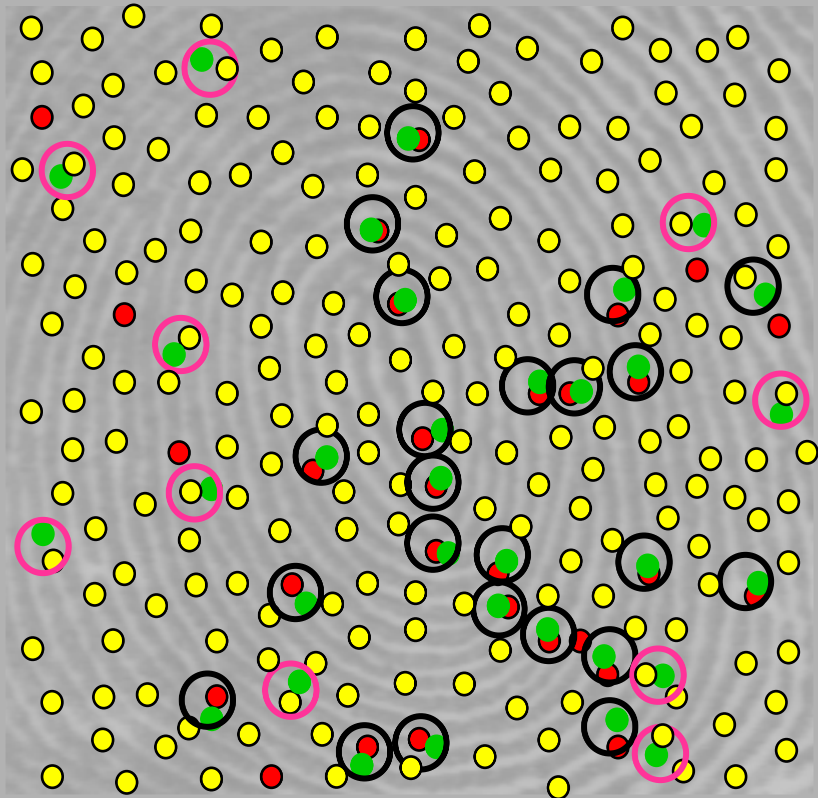
Given another point set **Y**, which is a noisy version of **X**, we can match each point to its nearest point in **R**

R



Now, list down the indices of the matched point in P_X:
5, 7,...

Out of 33 points in Y, 23 points matched to the original. The second part of the sketch is based on known techniques on set-different. It is designed in such a way that, if there is small number of miss-matched, the original can be recovered.

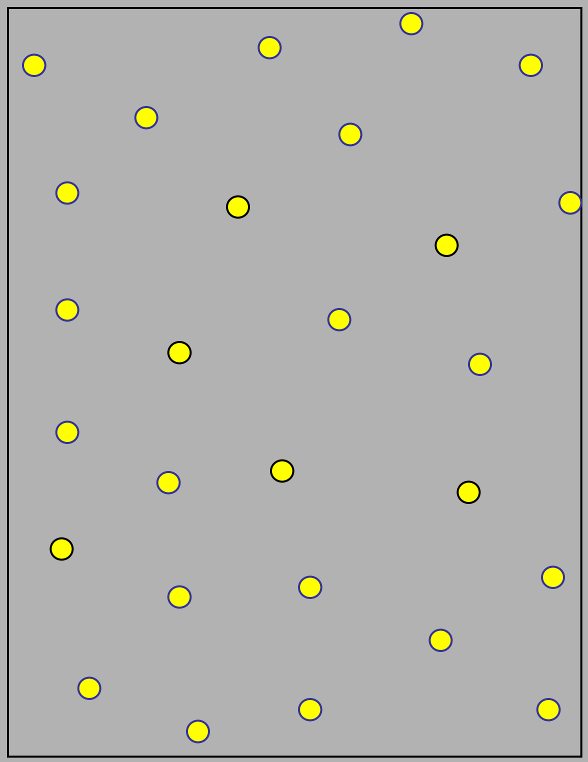


The problem

- In this paper, we are concern with the first sketch R. Given R, we want to investigate how much it reveals about X.

  In other words, given a well-separated R, we want to *guess* the original X.

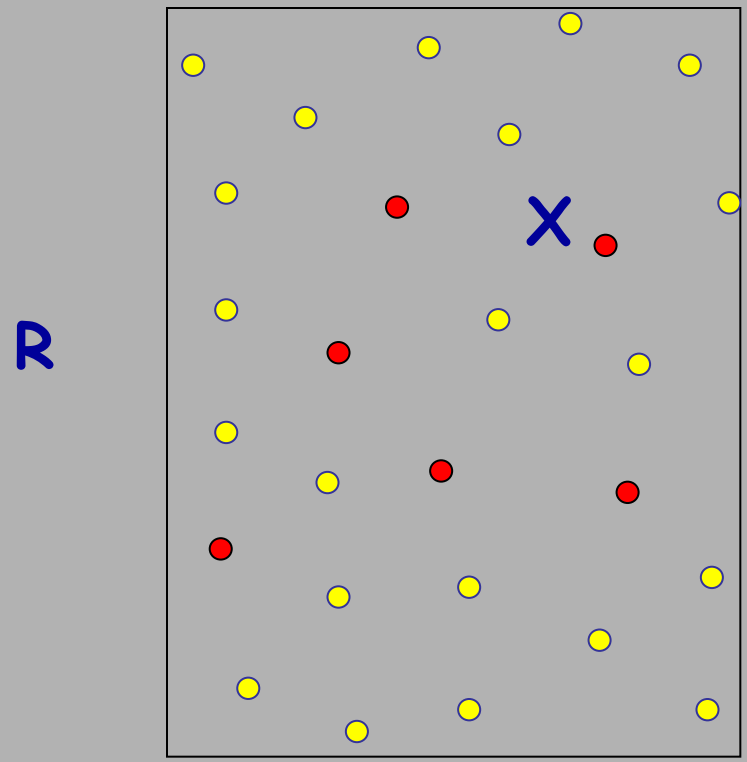
R



The problem

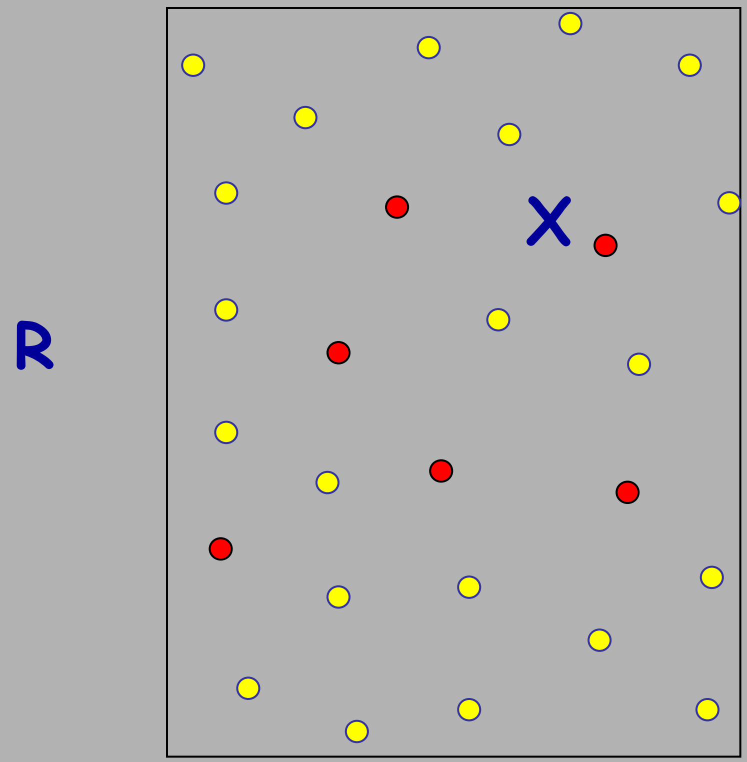
- In this paper, we are concern with the first sketch R. Given R, we want to investigate how much it reveals about X.

  In other words, given a well-separated R, we want to *guess* the original X.



The problem

- Assuming that the original X is also generated by the on-line parking process. Then R is generated by the online parking (starting from 0 point).
- Given R We want to guess the 6 earliest points.

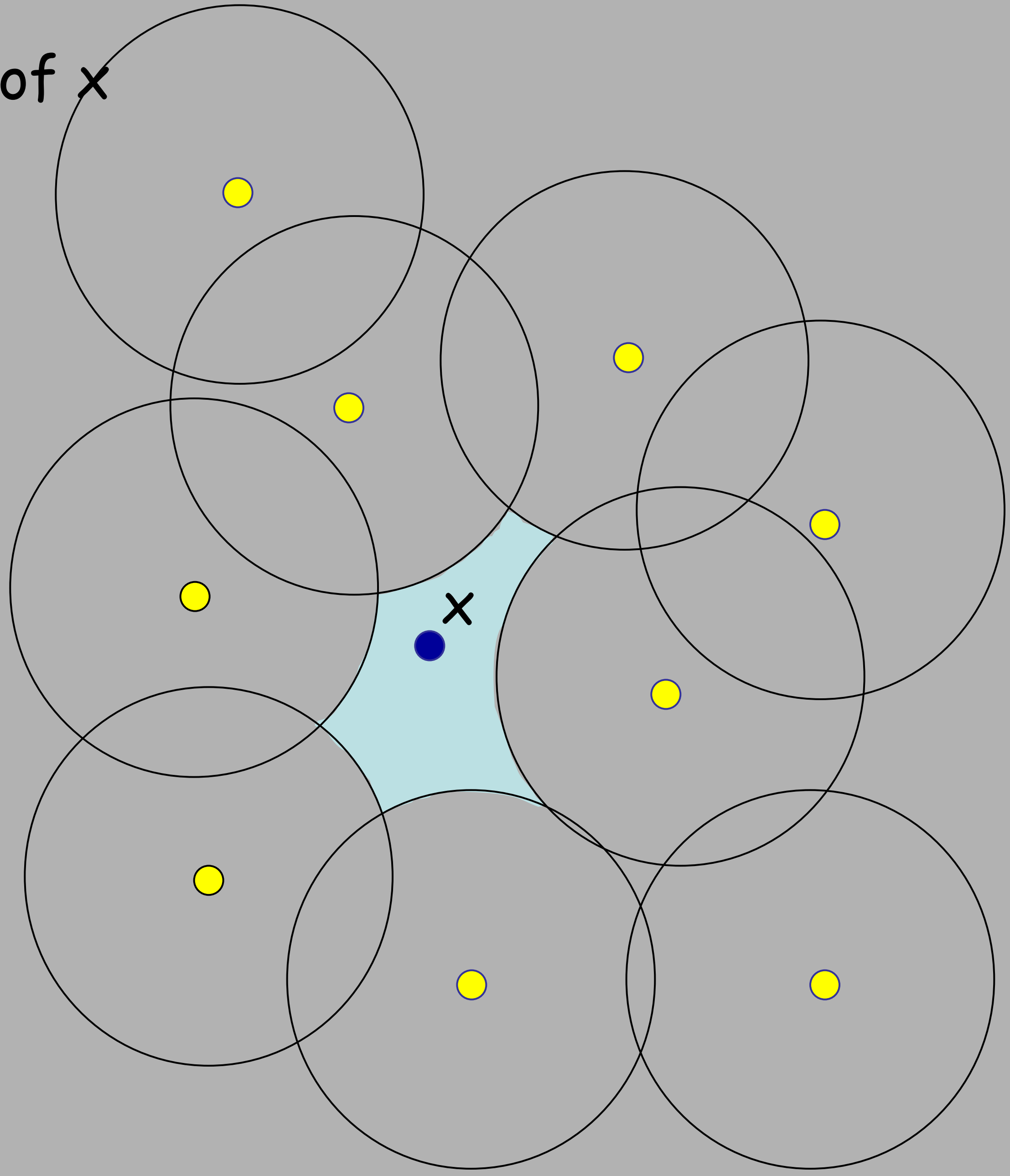


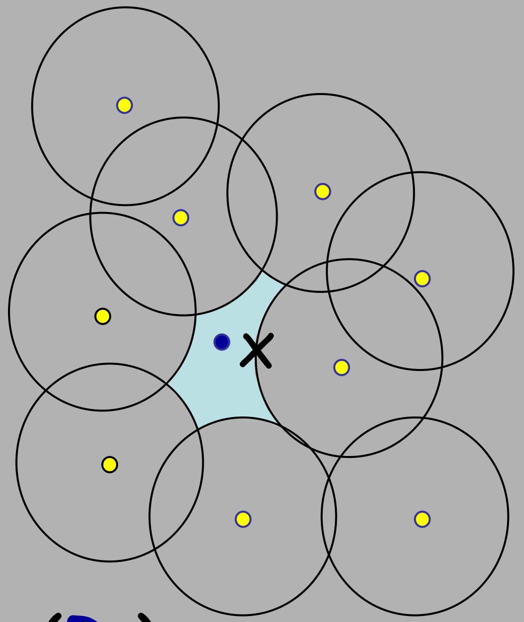
Some intuitions..

1) The online parking process is not memoryless. Hence, the statistical property of the *early-comers* "should" be different from the *latecomers*.

2) To distinguish *early* from *latecomers,* we probably should look into local neighbourhood of each point.

Free area of x
F(x)





Formally, w.r.t a point set **R**

Free_area of x = | Available_Region (**R**) –
Available_Region (**R**-{x}) |

where

Available_Region (R) is the region where we can add one more point s.t. the set remain well-separated.

Key Observation

- If $F(x) > F(y)$, then it is more likely that x arrives earlier than y.

- More formally, if $f_0 > f_1$, then for any s,

Pr (arrival order (x)<s | F(x) = f_0, x is selected) > Pr (arrival order (x)<s | F(x) = f_1, x is selected).

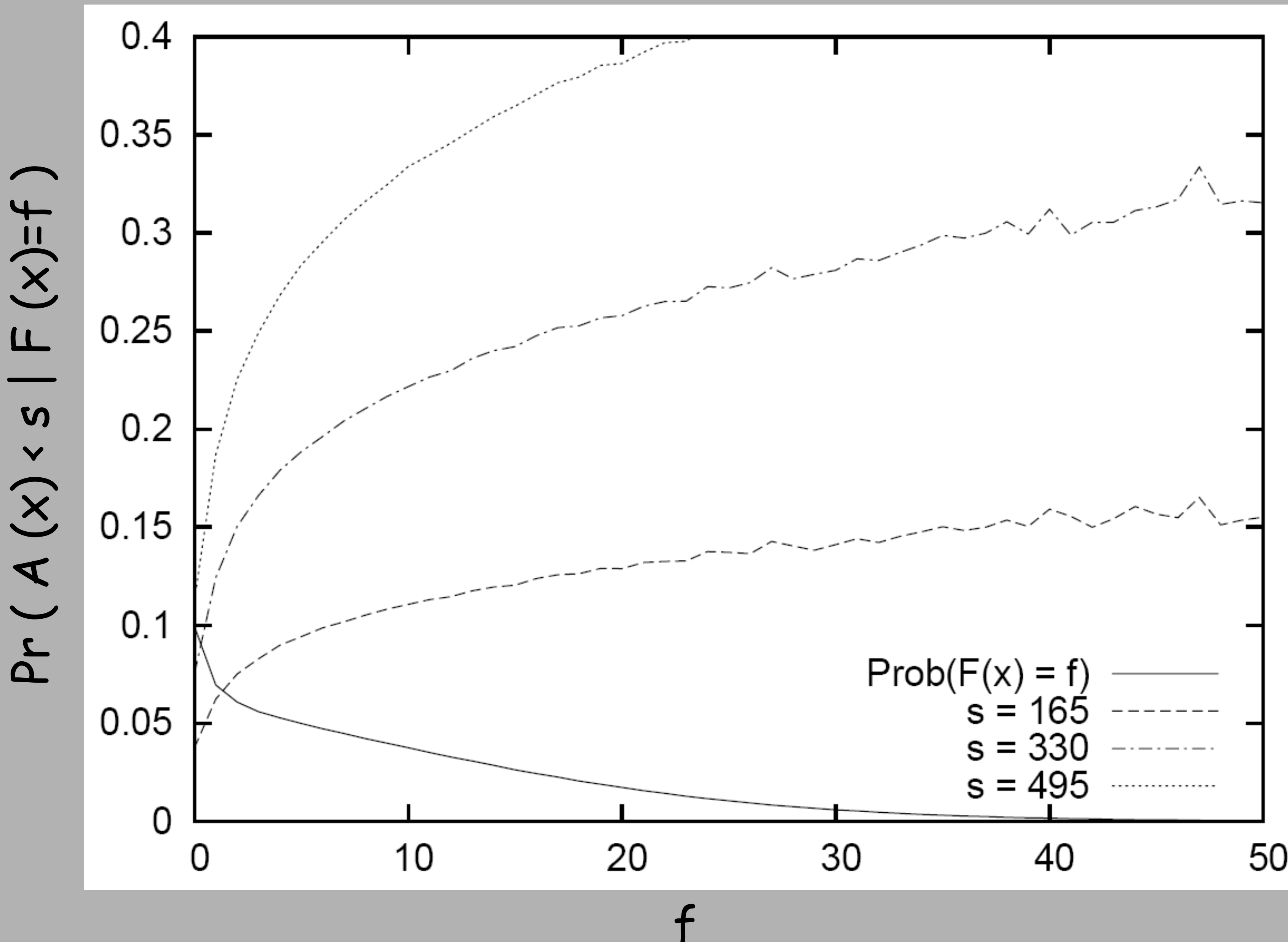
Key Observation

- If F (x) > F (y), then it is more likely that x arrives earlier than y.

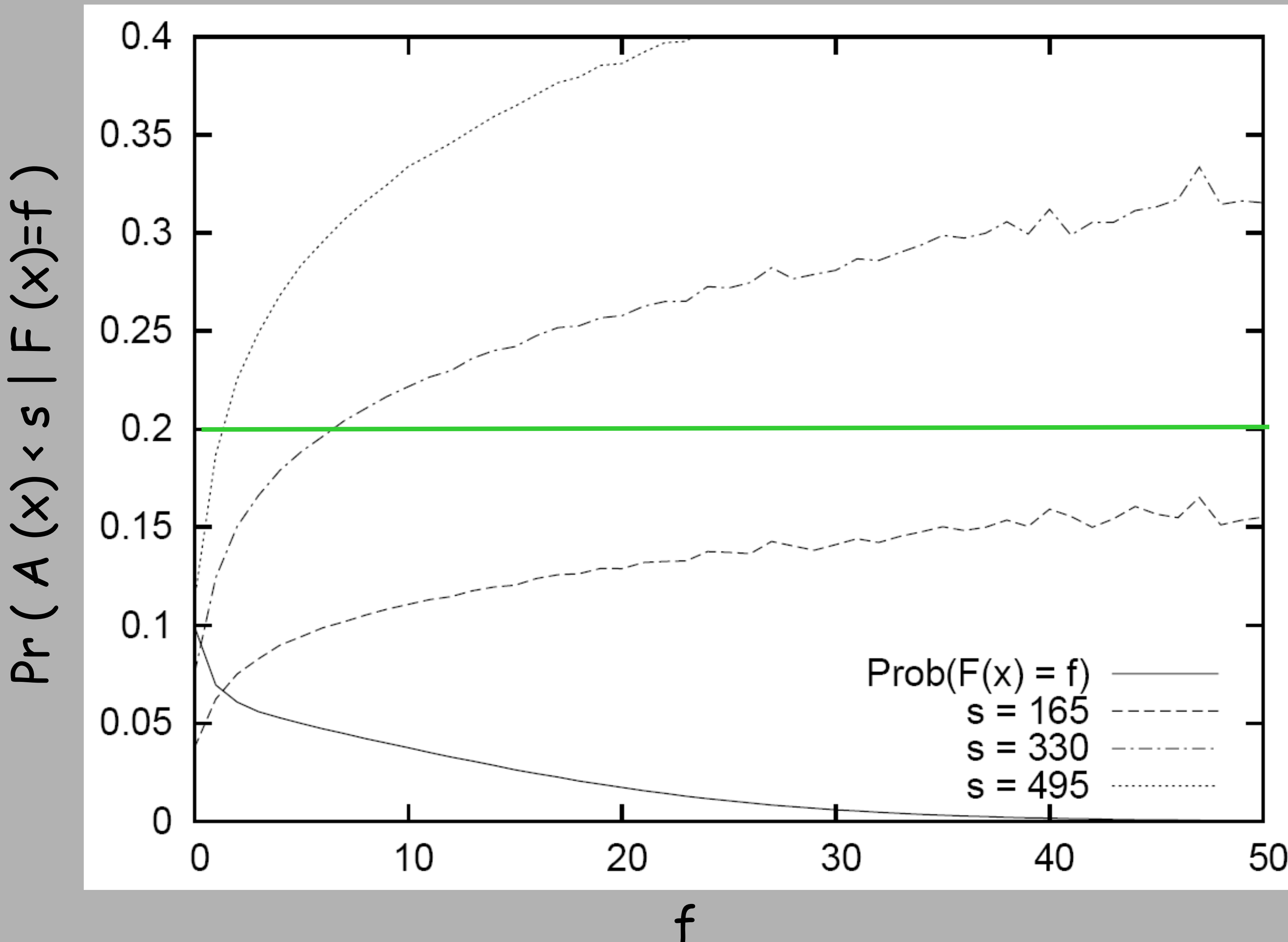
- More formally, if $f_0 > f_1$, then for any s,

  Pr (arrival order (x)<s | F(x) = f_0, x is selected) >
  Pr (arrival order (x)<s | F(x) = f_1, x is selected).

- The observation is verified through simulation. Unfortunately, we are unable to analytically prove it.

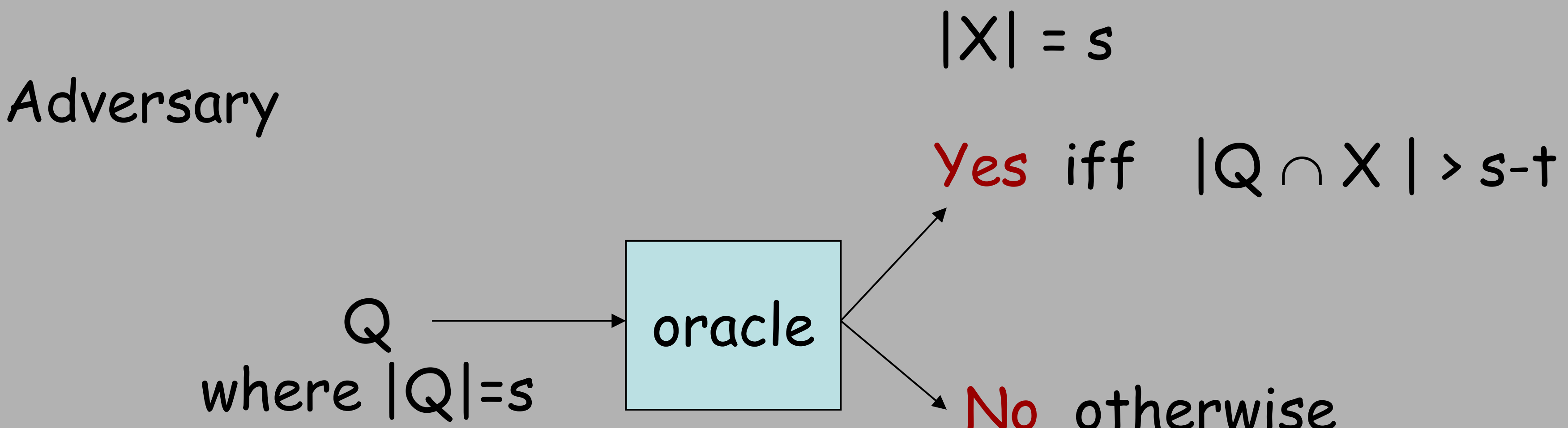


Total points ≈ 1668



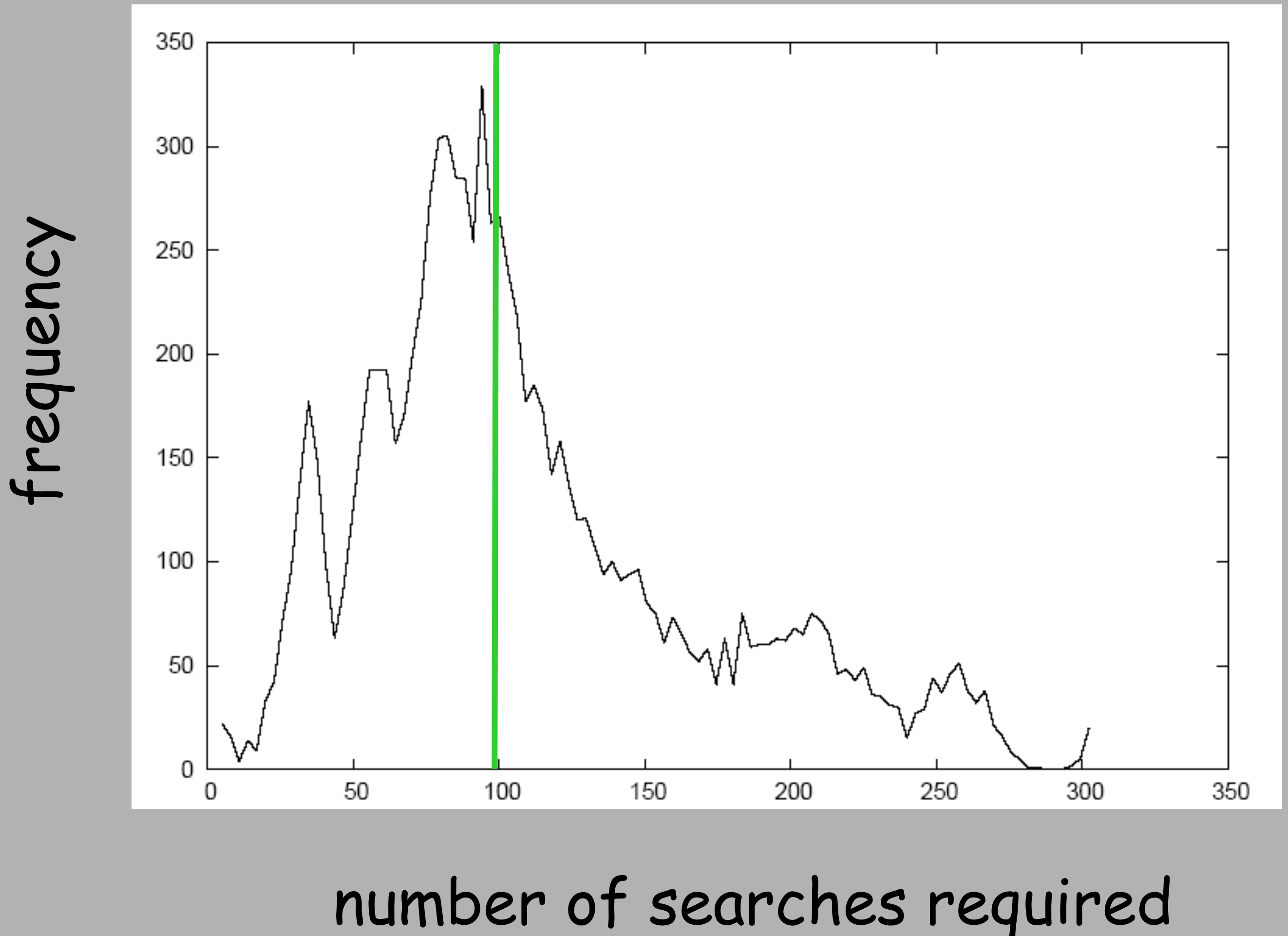
Total points ≈ 1668

Simulation: Model 1 – oracle attack



Simulation: Model 1 – oracle attack

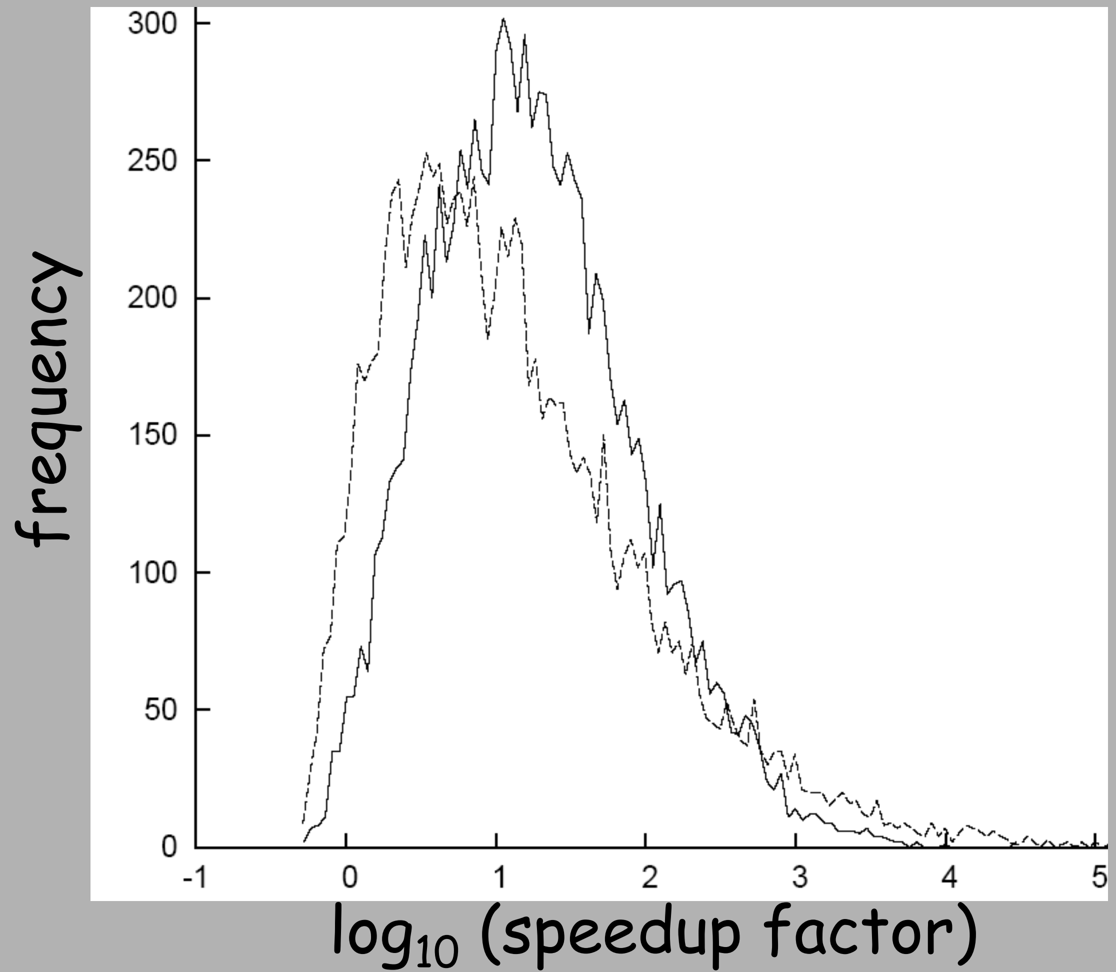
- When **s=1**, and the average total number of points is ≈ 318. The average number of searches required is ≈ 100.



Simulation: Model 1 – oracle attack

- When s = 38, average |R| ≈ 318

  average speedup compare to an exhaustive searches is ≈ 2192 times



Simulation: Model 2 – min entropy

We want to estimate,

$\log_2\{$

$-E[\ \max_X \Pr(\text{original point set} = X \mid \text{sketch} = R)]$

where expectation taken over the distribution of sketch.

$\}$

We can use the likelihood function to estimate an upper bound of the above. When s=38, and $|R| \approx 318$, the estimated upper bound is 61.2

$$\log_2 \binom{318}{38} > 150$$

Conclusion

- We show that the chaff points are not “random”.
- Entropy of point set is probably not too high. Although the speedup factor of our adversary is not overly large, it has to be taken into account to assess the security of a fingerprint sketch.
- Analytical proof - seems to be very difficult.

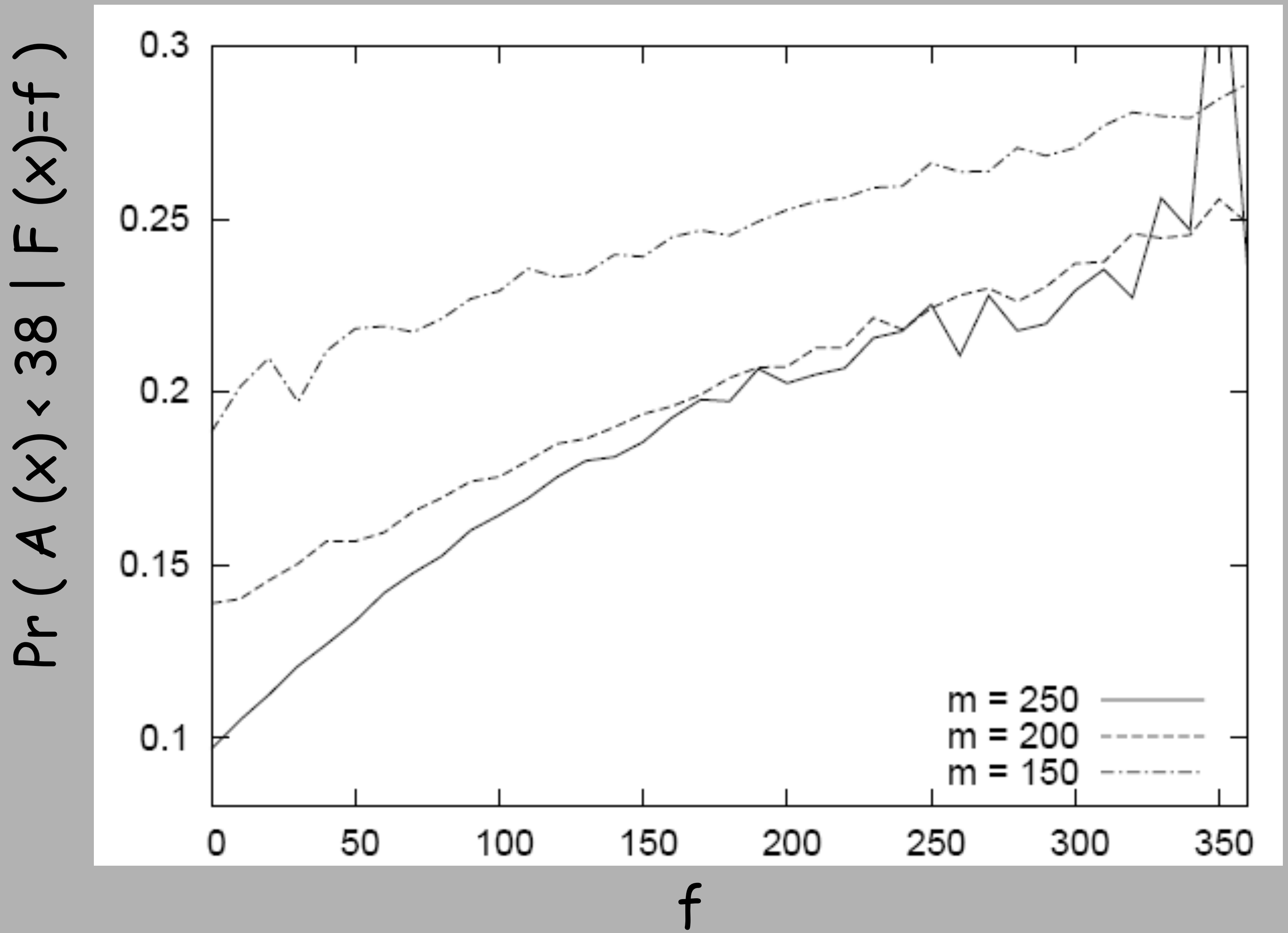
Future works

- Methods with "provable" bounds on entropy loss.

  Ee-Chien Chang & Qiming Li,
  Hiding Secret Points Amidst Chaff, Eurocrypt, 2006.

- Incorporating domain knowledge in the attack.

Online parking with fixed nos of points.



Approximation

- For more than one point,

$$Pr\ (\ A(x) < s_0,\ \ A(y) < s_1 \mid F(x) = f_0, F(y) = f_1) \ \approx$$

$$Pr\ (\ A(x) < s_0 \mid F(x) = f_1\ )\ .\ Pr\ (A(y) < s_0 \mid F(y) = f_1)$$

Background: Secure Sketch

- Biometric data is typically noisy. Two slightly different data may represent the same identity.

- This poses difficulties in applying classical cryptographic techniques, which are sensitive to small changes.