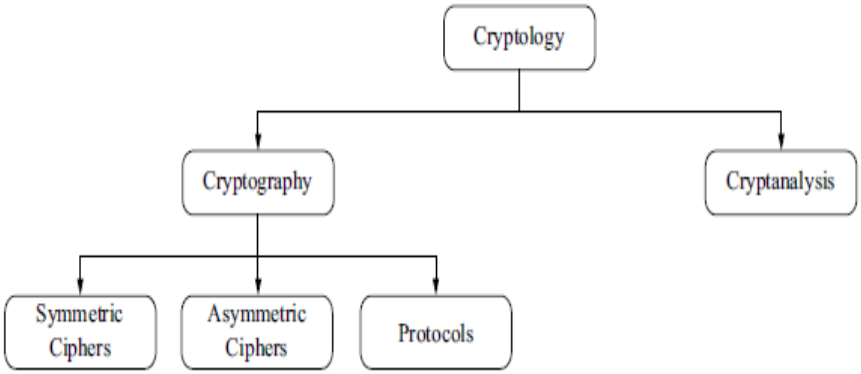
Machine leaning and Cryptography

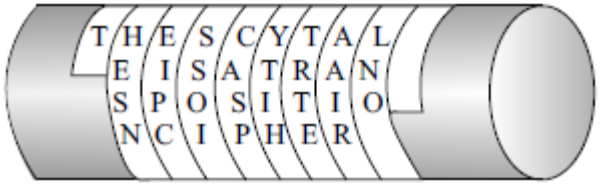
Cryptology

- **Cryptography** is the science of secret writing with the goal of hiding the meaning of a message.
- **Cryptanalysis** is the science and sometimes art of *breaking* cryptosystems



Enigma

Scytale of Sparta



Understanding Cryptography, Springer, 2010

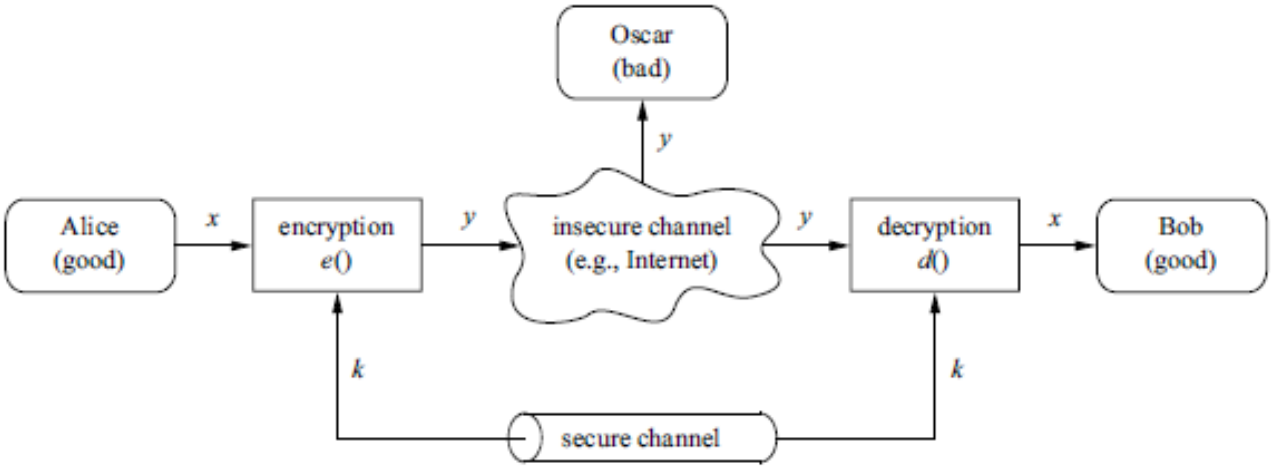


Fig. 1.5 Symmetric-key cryptosystem

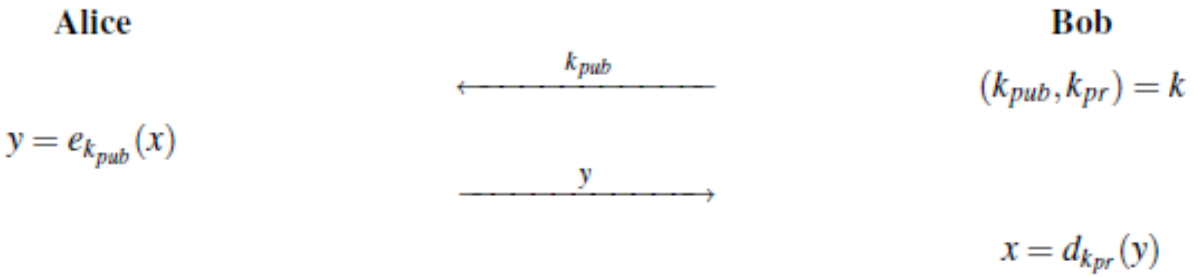
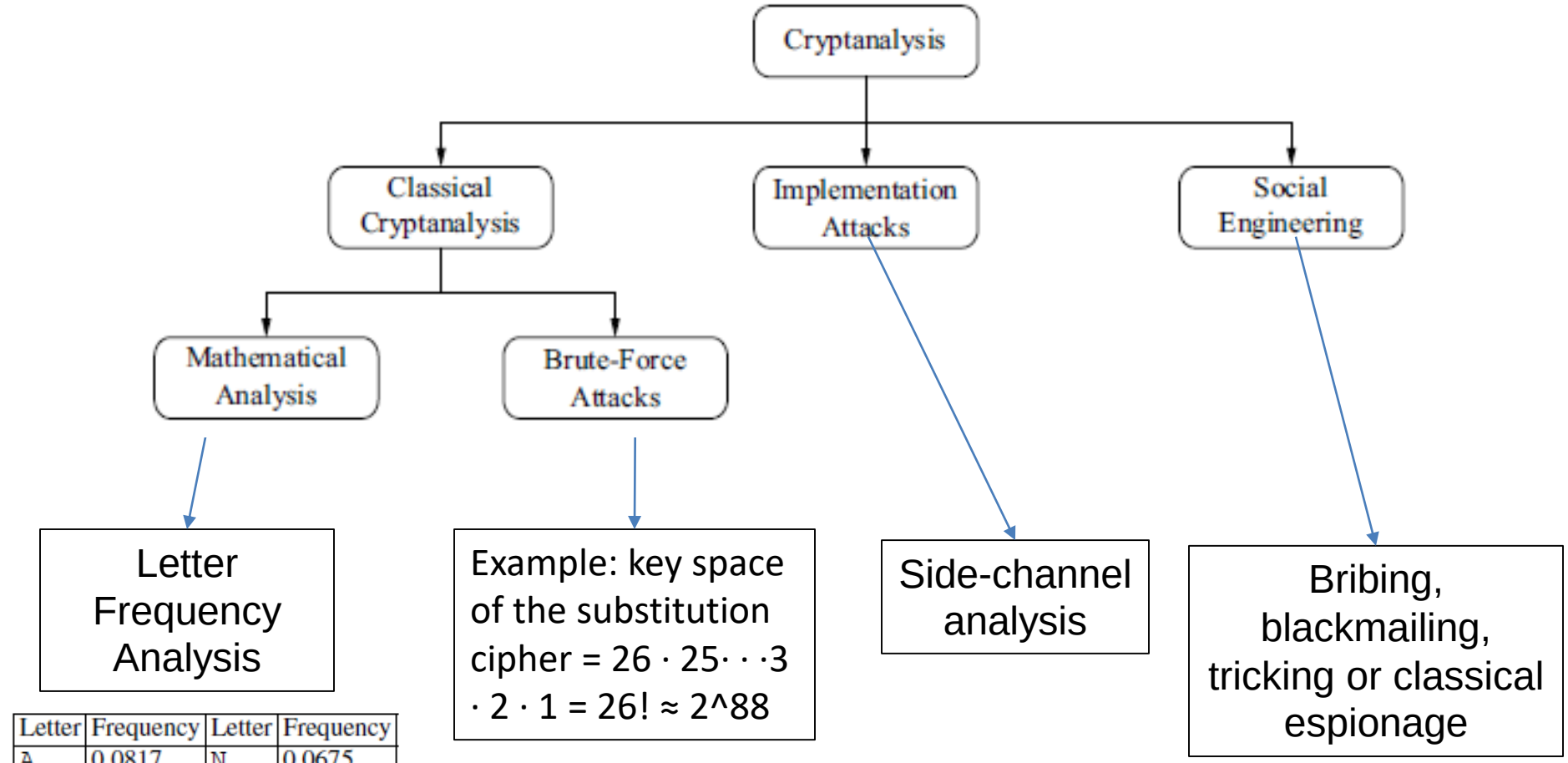


Fig. 6.4 Basic protocol for public-key encryption

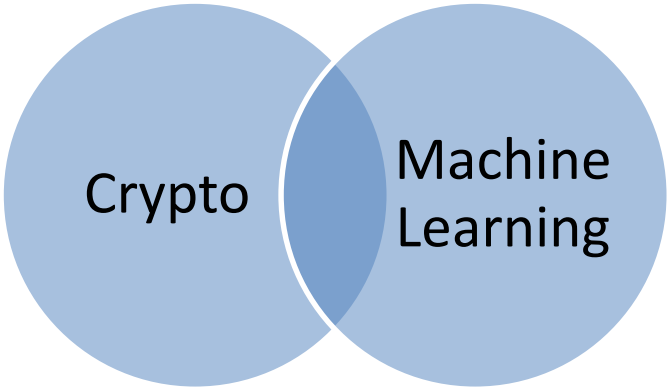


Letter	Frequency	Letter	Frequency
A	0.0817	N	0.0675
B	0.0150	O	0.0751
C	0.0278	P	0.0193
D	0.0425	Q	0.0010
E	0.1270	R	0.0599
F	0.0223	S	0.0633
G	0.0202	T	0.0906
H	0.0609	U	0.0276
I	0.0697	V	0.0098
J	0.0015	W	0.0236
K	0.0077	X	0.0015
L	0.0403	Y	0.0197
M	0.0241	Z	0.0007

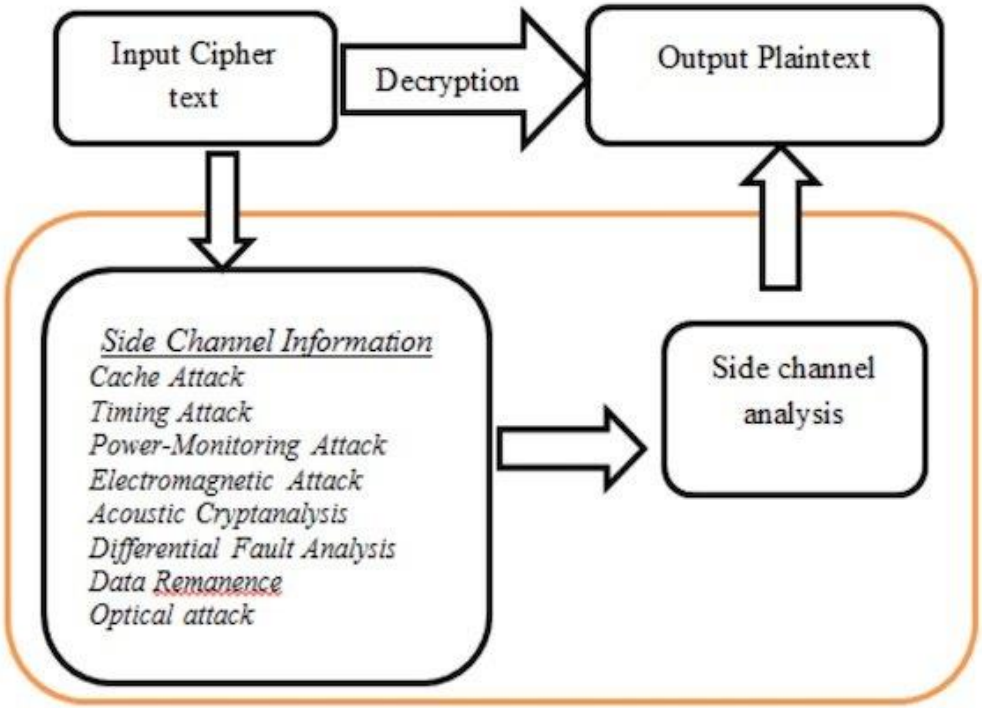
Side-channel analysis

- Side-channel analysis can be used to obtain a secret key, for instance, by measuring the electrical *power consumption* of a processor which operates on the secret key.
- The *power trace* can then be used to recover the key by applying signal processing techniques.
- In addition to power consumption, *electromagnetic radiation* or the *runtime behavior* of algorithms can give information about the secret key and are, thus, useful side channels.

Side-channel analysis



- **ML for Cryptography**
  - Side channel analysis attacks

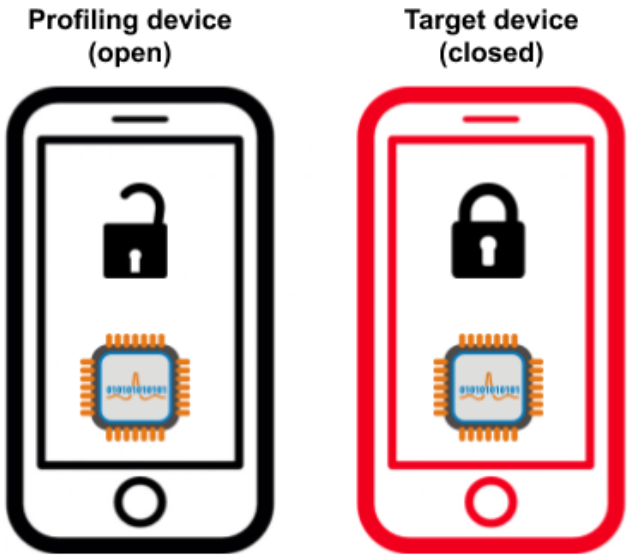


SCA Attacks

Non Profiled attacks

- Differential Power Analysis (DPA)
- Correlation Power Analysis (CPA)
- Mutual Information Analysis (MIA)

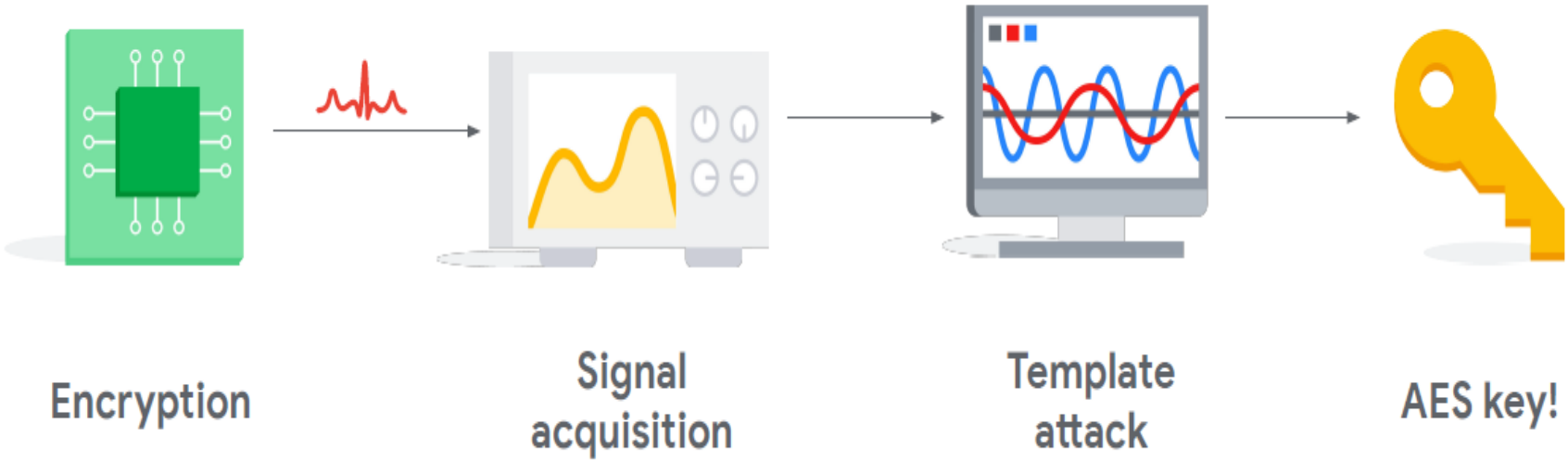
Profiled attacks



- Template attacks
- Support Vector Machine
- Random Forests
- Deep Learning

SCA Attacks

- SCA attacks are passive and non-invasive
- Profiled SCA: profiling and attacking



https://elie.net/talk/a-hackerguide-to-deep-learning-based-side-channel-attacks/

Power monitoring attack

- Because the amount of power used is related to the data being processed, power consumption measurements contain information about the circuit's calculations
- When a device is processing cryptographic secrets, its data-dependent power usage can expose those secrets to attack.
- A power SCA attack typically requires the insertion of a small *resistor* (0.5–10 Ω) in series with the power supply of the device that measures the voltage drop across it

https://doi.org/10.3390/cryptography4020015

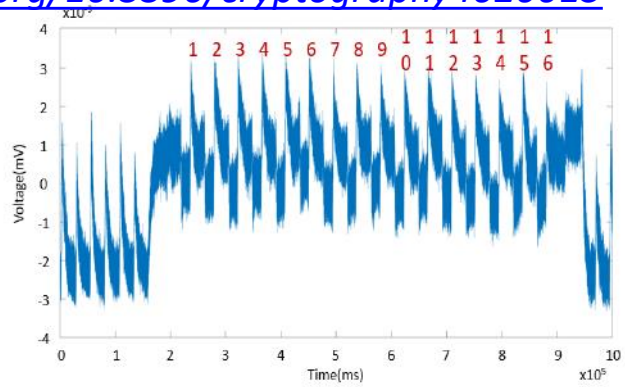


Figure 1. Simple Power Analysis (SPA) trace showing an entire Digital Encryption Standard (DES) operation.

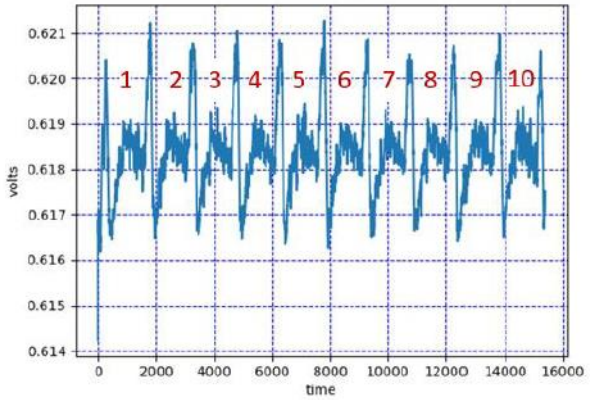
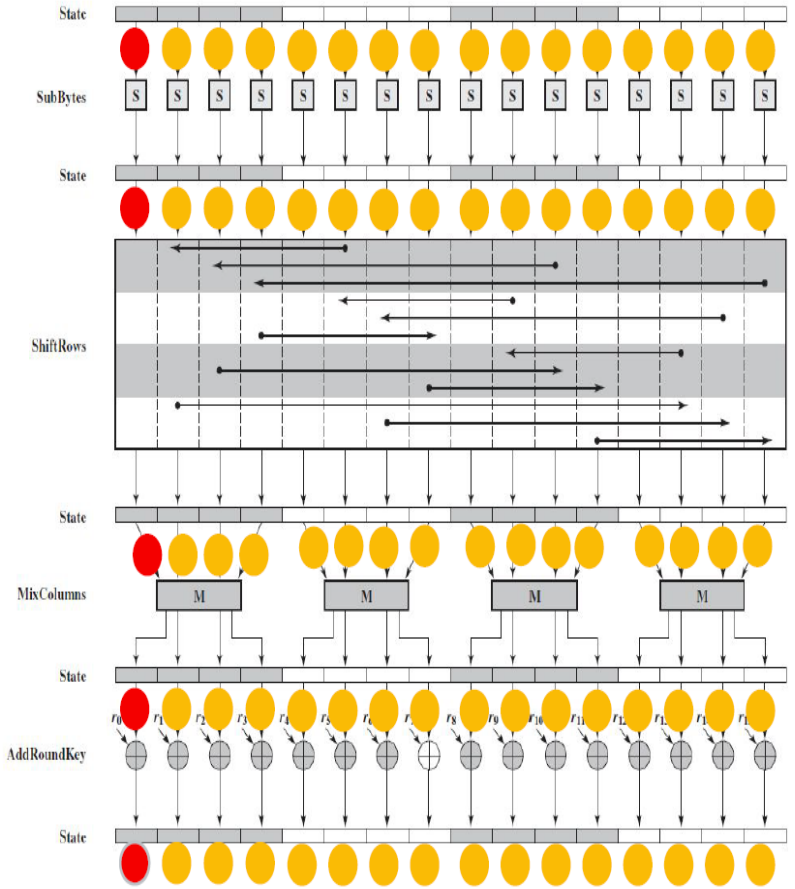


Figure 2. Simple Power Analysis (SPA) trace showing an entire Advanced Encryption Standard (AES) operation.

Each collection of power measurements taken over a period of interest (often a full cryptographic operation) is referred to as a trace



https://elie.net/talk/a-hackerguide-to-deep-learning-based-side-channel-attacks/

Power monitoring attack

- To recover an *n*-bit key $K \in K\_key$, where K is the set of all possible keys, the attacker uses a set *P* of known input data (e.g. the plaintext) and a set *T* of physical measurements (e.g. power consumption).
- Typically a divide-and-conquer strategy is applied in which the key *K* is divided into *b*-bit parts K_k, called subkeys, and the subkeys are recovered independently, for $k \in \{1,2,\ldots, n/b\}$.
  - Typically the size of the subkey is a byte, $b = 8$.
- After the attack, the attacker gets *n/b* vectors of probabilities p_k, in which the element $p_{k,j}$ represents the probability that the subkey $K_k = j$ is the correct subkey, for $j \in \{0,1,\ldots,2^b - 1\}$

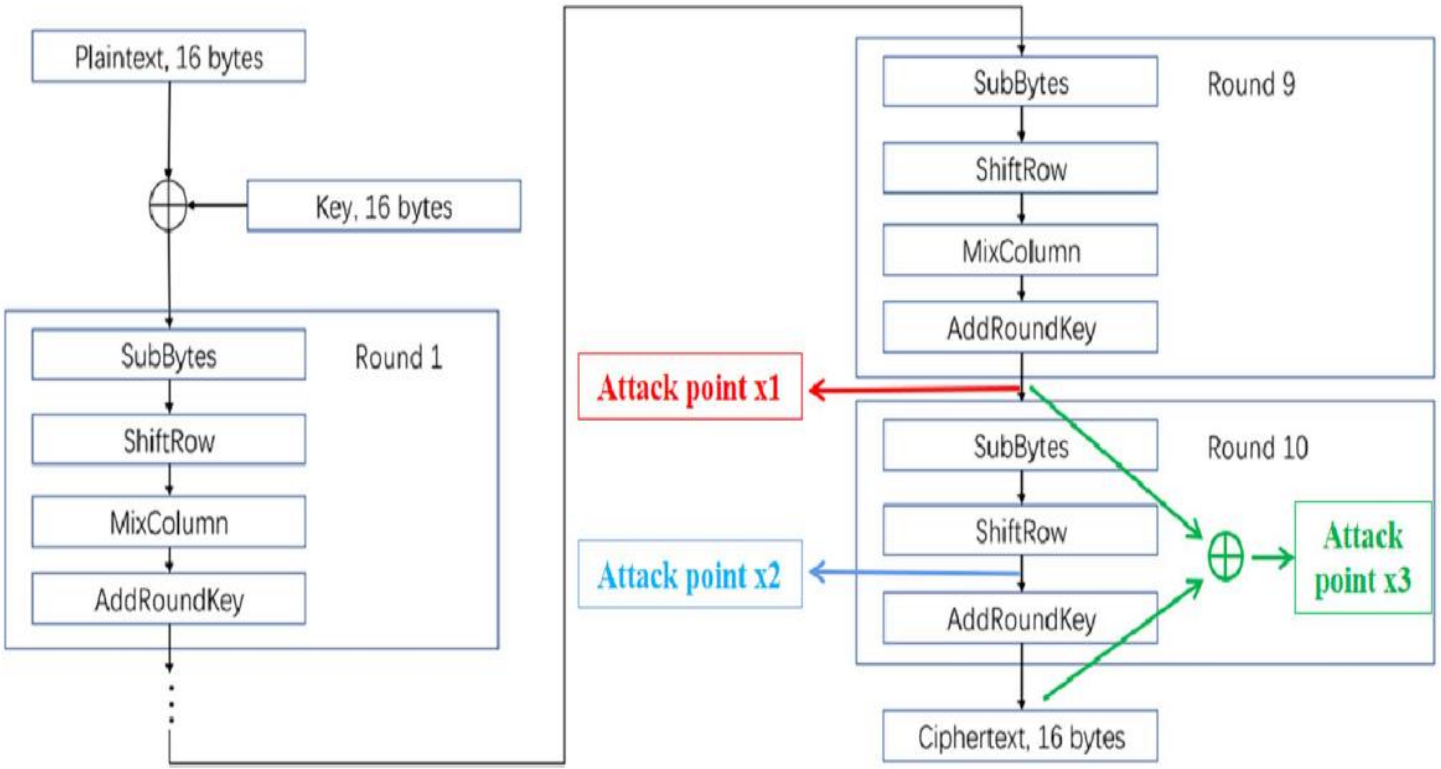
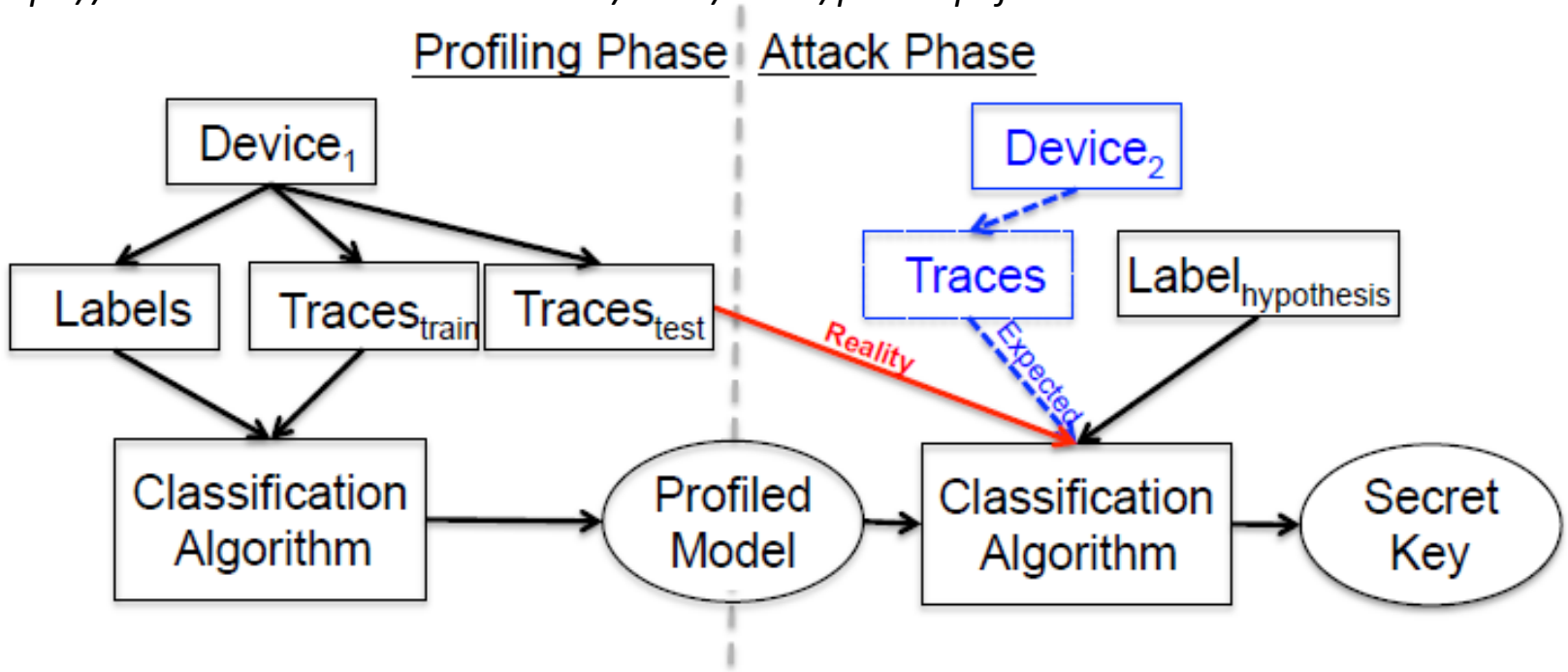


Fig. 1 AES-128 algorithm and three attack points we used to train CNN models [26]

- AES-128 is a symmetric encryption algorithm, which takes a 128-bit block of plaintext and a 128-bit key as inputs.
  - Figure 1 shows the flow of the AES-128 algorithm and attack points (i.e three used here).
- AES-128 contains 10 rounds in total.
- Except for the last round, each round has 4 steps: SubBytes, ShiftRows, MixColumns and AddRound-Key.
- The last round does not mix columns. The SubBytes procedure is a byte-to-byte substitution using a lookup table called *Substitution Box* (SBox).

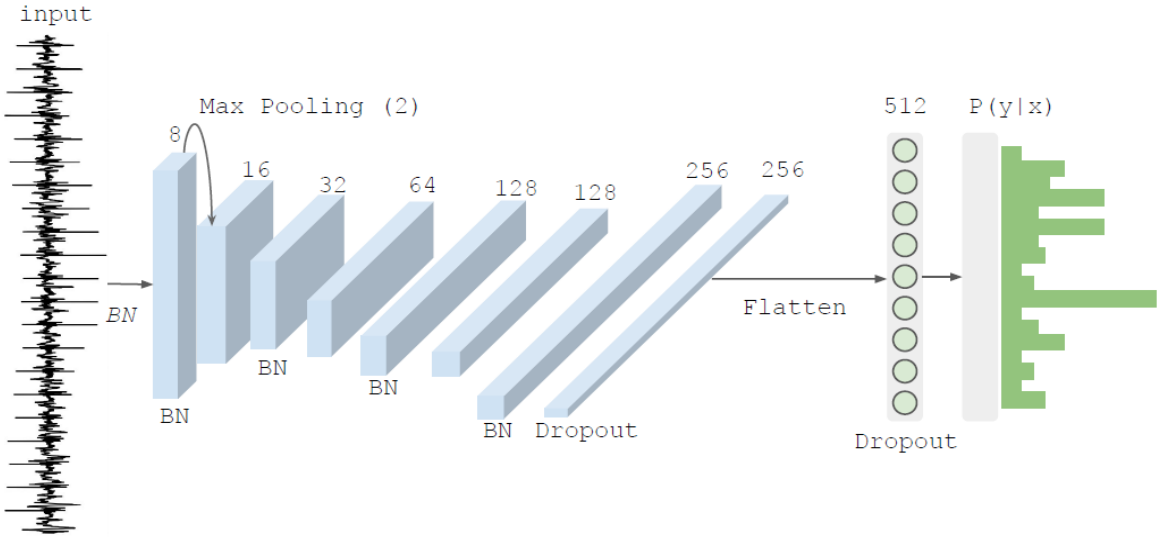
https://doi.org/10.1007/s42979-021-00755-w

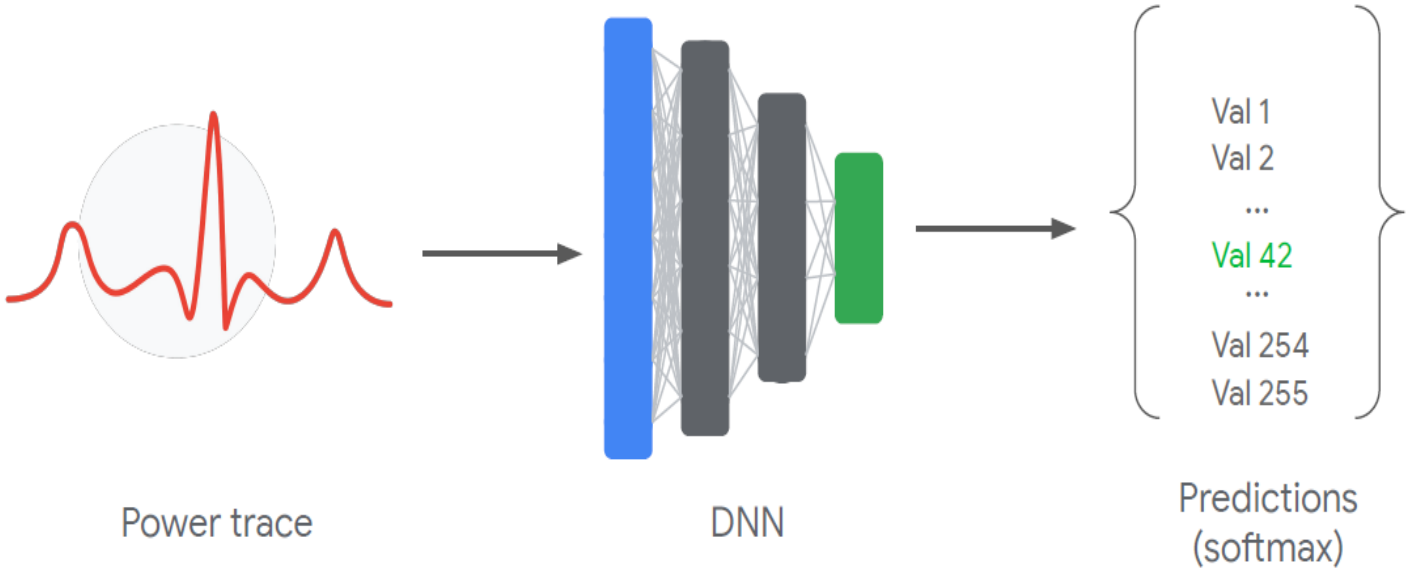
https://summerschool-croatia.cs.ru.nl/2019/slides/picek1.pdf



- Assumptions. Profiled side-channel attacks typically assume that:
  - (1) The attacker has a device(s), called the profiling device, which is similar to the device under attack, called the target device.
  - (2) The attacker has a full control over the profiling device.
  - (3) The attacker has a direct physical access to the target device for a limited time.

https://summerschool-croatia.cs.ru.nl/2019/slides/picek1.pdf





https://elie.net/talk/a-hackerguide-to-deep-learning-based-side-channel-attacks/

Profiling stage

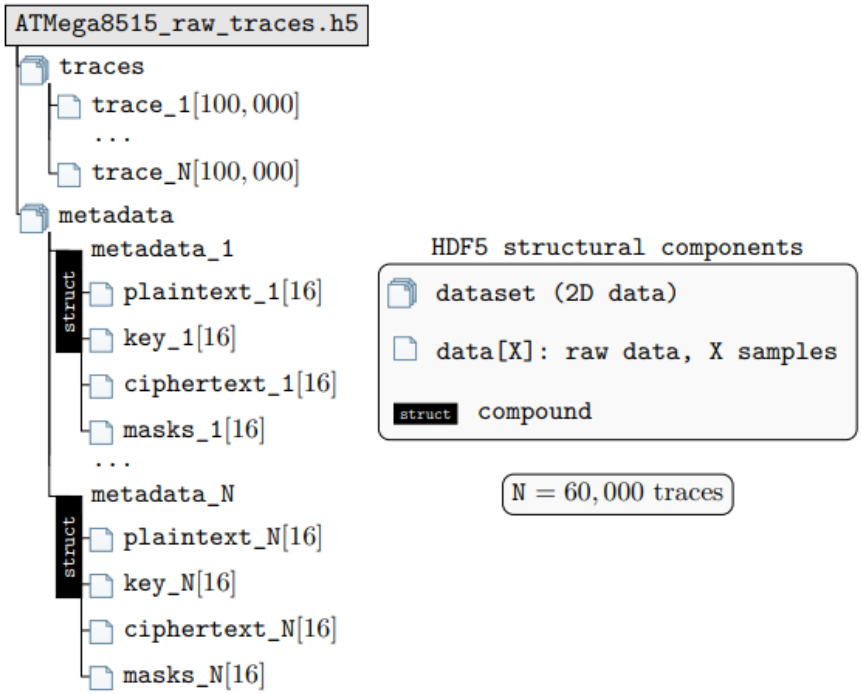
- At the profiling stage, the attacker first uses the profiling device to encrypt a large number of plaintexts using known keys and captures traces.
- The model is trained on the labeled traces to learn the correlation between traces and keys.
- The traces labelling: the number of labels is the total of all possible intermediate values at an attack point.

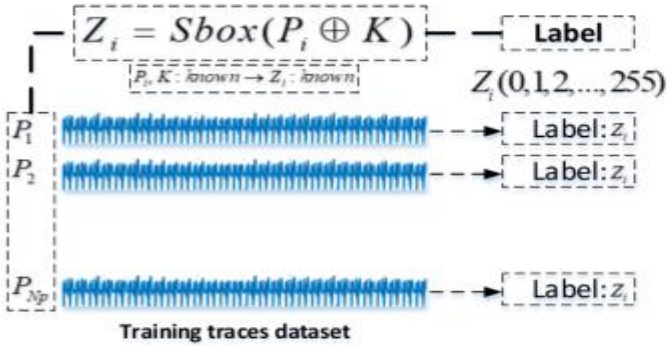
Attack stage

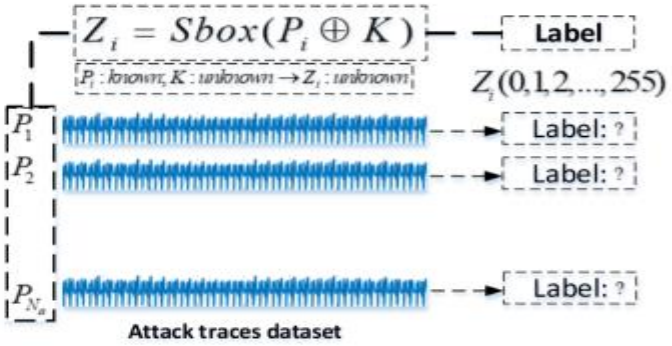
- At the attack stage, the attacker uses the victim device to encrypt a small number of plaintexts and records corresponding traces.
- Using the trained model to classify traces captured from the victim device, the attacker is able to obtain the corresponding intermediate data and hence derive the subkey.

ASCAD database

- Like MNIST database
- Format: traces consist of 100000 time samples measured during the masking AES-128 running on smartcard
- Size: 60000 (50000 for profiling, 10000 for testing)
- Label: the number of labels is the total of all possible intermediate values at the output of sbox of first round.







Cryptography and Machine Learning

- Crypto for ML
  - Security for training data and model parameters: Data encrypted

Encrypted computation

- Encrypted computation is a subfield of cryptography that allows you to compute on encrypted data without decrypting it
- Encrypted computation solves a security issue when multiple parties want to compute together but don’t want to compute using unencrypted, or plaintext, data

When to Use Encrypted Computation

- Insecure cloud environments
- Computing with third parties
- Computing on encrypted data
- Comparing data across parties
- Distributing data or secrets
- Guaranteeing collaboration and distributing control
- Liability

Privacy Versus Secrecy

- Privacy is the guarantee that information about an individual is not leaked or revealed without their consent or awareness.
- Privacy in cryptography is the ability to decide and guarantee what data is revealed to whom.
  - There are no assumptions about the leakage of that information once it is revealed.
  - This property is also called *secrecy* because it communicates that the values remain secret so long as they are encrypted

Secure multiparty computation

- Secure multiparty computation (often SMPC or MPC for short) is a subfield of encrypted computation that involves multiple parties agreeing to compute together but only if the data is encrypted.
- The typical steps of an MPC scheme include encrypting the input data via clever mathematical transformations and then sharing the encrypted data across multiple parties.
- The parties then cooperate and exchange intermediary values until, ultimately,
  - the final result is revealed via a set of decryption mechanisms and interactions.

Secure multiparty computation

- MPC provides interesting secrecy properties.
  - The parties involved in the computation would like to keep their inputs secret and ensure no other player learns much about their input information.
- Secret Sharing example:

```
x = 45
keys = [100, 22, 43, 56]
enc_x = x - sum(keys) # enc_x is -176

# to "decrypt" you can add the keys to the "encrypted" value
enc_x + sum(keys)
```

a curious participant could determine information about the size of x (i.e., whether it is a large or small number)

```
Q = 431
x = 45
keys = [100, 22, 43, 56]
enc_x = (x - sum(keys)) % Q ❶
```

enc_x is 255 instead of –176, guess?

HE

- HE, like MPC, computes results without ever decrypting the data.
- HE does this via special cryptosystems that maintain homomorphic properties, allowing encrypted arithmetic operations on the data.
- Homomorphism is the ability to map algebraic structures, such as groups or vectors, to one another while maintaining the mathematical properties for certain operations, like addition or multiplication.
- In this case, the ciphertext and the plain text are homomorphic, allowing you to accurately perform mathematical operations on ciphertext.

HE

- Example:
- The ElGamal cryptosystem is additively and multiplicatively homomorphic.
- This means that there exist two operations over the ciphertexts $E(M_1)$ and $E(M_2)$ such that the results of those operations correspond to new ciphertexts whose decryption yield the sum and multiplication of the plain texts M_1 and M_2.

$$E(M_1).E(M_2) = E(M_1 + M_2)$$

$$E(M_1)^{M2} = E(M_1.M_2)$$

Recommender system

- There are two basic entities that drive a recommender system:
  - users who use the recommender system to provide opinions as well as receive recommendations and items that are rated by users.
- The inputs to a recommender system are usually arithmetic rating values, which express the users' opinions of items and follow a specified numerical scale (example: 1: bad to 5: excellent).
- The outputs of a recommender system can be either predictions or recommendations

Recommender system

- Let U = u1; u2; . . .; u_n be the set of all n users in a recommender system and I = i1; i2; . . .; i_m be the set of items where *m* is the total number of items.
- Let R be a rating matrix where $r_{i,j}$ is a rating provided by user u_i on item i_j.
- To generate recommendations, one of the key steps is to calculate similarities/correlations among the item pairs.
  - Cosine similarity is one of the commonly adopted similarity measures to determine the nearest neighbour in recommendation generation

$$s(i_j, i_k) = \frac{\sum_{i=1}^{n} r_{i,j} r_{i,k}}{\sqrt{r_{1,j}^2 + \cdots + r_{n,j}^2}\sqrt{r_{1,k}^2 + \cdots + r_{n,k}^2}} \qquad (1)$$

where i_j and i_k represent two individual items. $r_{i,j}$ and $r_{i,k}$ represent the ratings provided by the user u_i on those two items and n represents the total number of users.

Table 2 User-item rating matrix

Users/items	i_1	i_2	...	...	...	...	i_m
u_1	$r_{1,1}$	...	$r_{1,j}$	...	$r_{1,k}$	...	$r_{1,m}$
u_2	$r_{2,1}$	...	$r_{2,j}$	...	$r_{2,k}$	...	$r_{2,m}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
u_i	$r_{i,1}$	...	$r_{i,j}$	...	$r_{i,k}$	...	$r_{i,m}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
u_n	$r_{n,1}$	...	$r_{n,j}$	...	$r_{n,k}$	...	$r_{n,m}$

CBF-Based Recommendations

- In CBF, the recommendations are generated based on the items' features.
- The process is to check for similarity among the items which is calculated using item features first, and then, based on those similarity, the CBF generates recommendations for the target user

$$P_{i,k} = \frac{\sum_{j=1}^{m} r_{i,j} \cdot s(i_j, i_k)}{\sum_{j=1}^{m} s(i_j, i_k)} \quad (2)$$

where $P_{i,k}$ denotes the rating prediction for user u_i. $k = \{1, 2, \ldots, m\}$ is the number of items that the target user has requested for the recommendation. $r_{i,j}$ and $s(i_j, i_k)$ denote the rating vector of user u_i and similarity between item i_j and i_k respectively.

The need to calculate average values without revealing rating from users

Privacy-Preserving Recommender System

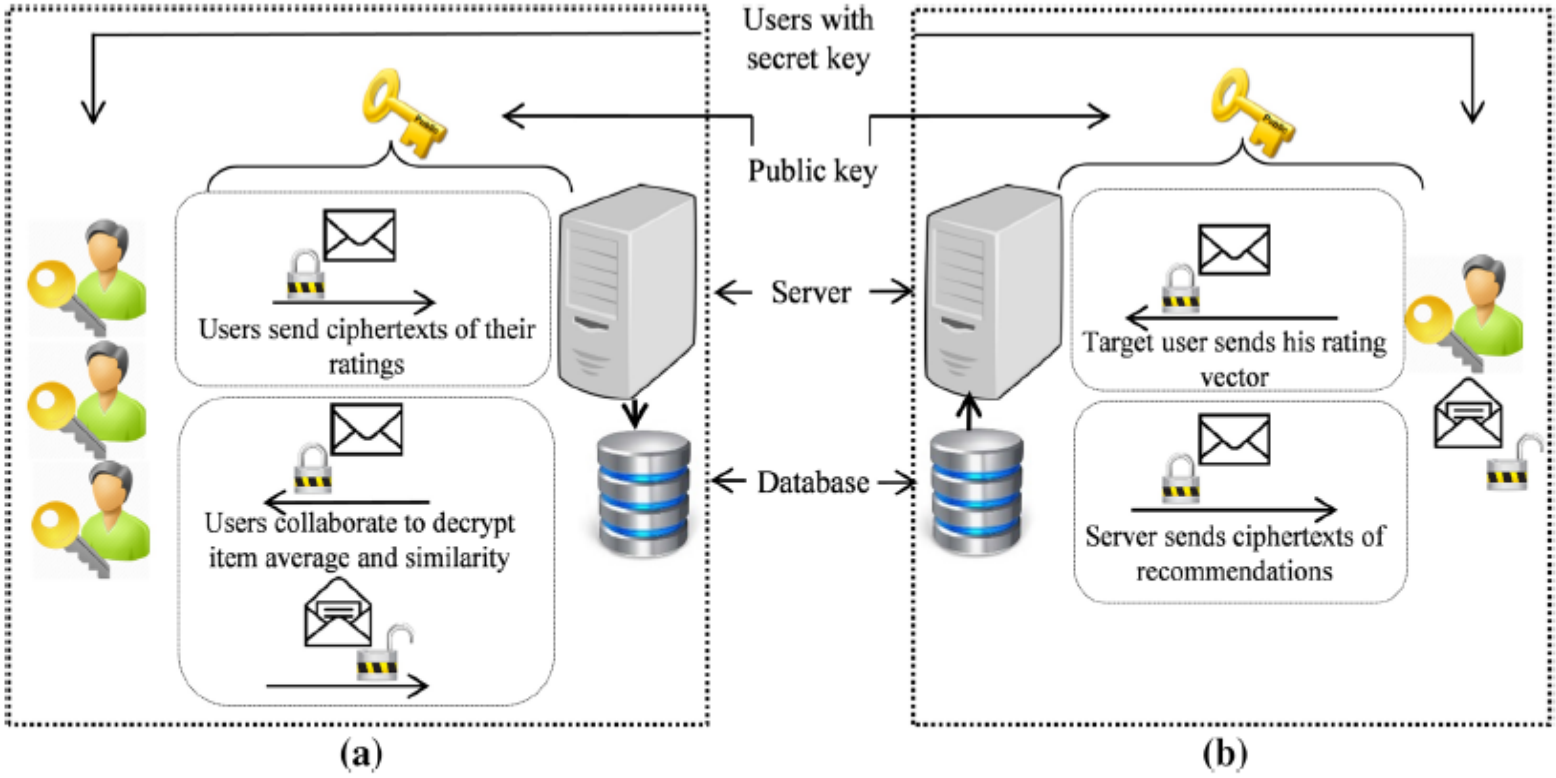


Fig. 2 Framework of proposed privacy-preserving recommender system, divided into two phases: **a** *average and similarity computation*, all users participate and send the ciphertexts of their ratings to the server. Server performs homomorphic operation to calculate average and similarity, thereby stores the results in its own database, **b** *recommendations generation*, only one user participates and send the ciphertexts of own ratings. The server computes the recommendation using homomorphic properties and send the ciphertexts to the user. The decrypts the ciphertexts using own private key

DOI 10.1007/s41019-016-0020-2

Conclusion

- SCA attacks and be simulated by machine learning.
- SMC can be used for privacy preserving machine learning