SiPS: October 21, 2013 - Belfast

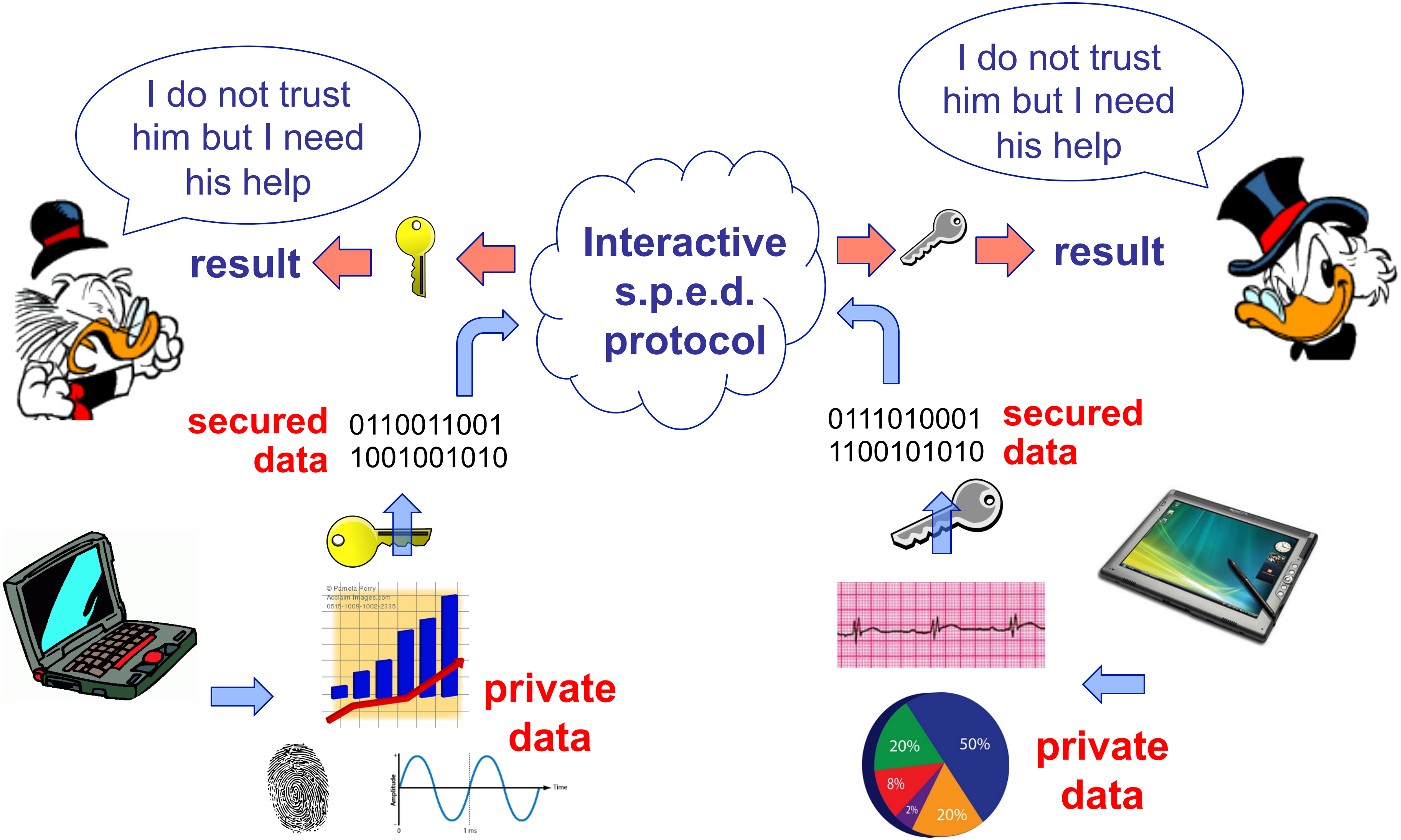
# Computing with private data: when cryptography meets signal processing

**Mauro Barni**
University of Siena

# Outline

- **What** am I talking about and **why** it is interesting ?
- **How** does it work ?
- **s.p.e.d.** at work: the SP side of the coin

# What: the s.p.e.d. paradigm



I do not trust him but I need his help

I do not trust him but I need his help

**Interactive s.p.e.d. protocol**

**result**

**result**

**secured data** 0110011001 1001001010

0111010001 1100101010 **secured data**

**private data**

**private data**

# Why? Network and web security

- Privacy-Preserving Intrusion Detection
  - Analysis of private log files, traffic monitoring
- Abuse detection in social networks
  - Chat rooms or messaging services ensure user anonymity
  - Users should be traceable if they severely violate the terms of usage.
  - To limit traceability to severe instances, abuse detection could be carried out on encrypted data and anonymity revoked only in case of violation
- Oblivious Web Ratings
  - The popularity of web pages is assessed by a third party analyzing the encrypted log files of a web server

# Why ? Profiling / recommendation services

- Targeted Recommendations
  - Personalized recommendations have high business value but open a privacy-problem
  - Problems can be avoided by methods that analyze the relevant user habits in the encrypted domain.
- Data Mining for Marketing
  - Knowledge of preferences of class of users is invaluable information in marketing.
  - Performing classifications in the encrypted domain can prevent privacy concerns

# Why ? Access control and biometrics

- Private Access control via encrypted queries
  - Access to a service is granted upon inspection of a biometric template (BT)
  - The BT is encrypted so to avoid revealing the biometry and the identity of the user accessing the service
- Biometric control in public places (airport …)
  - An encrypted BT is used to look for criminals or terrorists in public locations
  - Only if a match is found the identity is revealed thus avoiding tracing honest citizens

# Why ? Biomedical data processing

- Storing biomedical data on remote servers
  - Medical sensitive data/signals are stored under encryption
  - Additional services are provided by processing the encrypted data
  - Google-health
- Privacy-preserving remote services
  - a remote diagnosis services analyses encrypted data and provides recommendations without violating the users' privacy
- Analysis of bio-signals
  - by processing encrypted bio-signals the analysis reveals only the information it is intended for

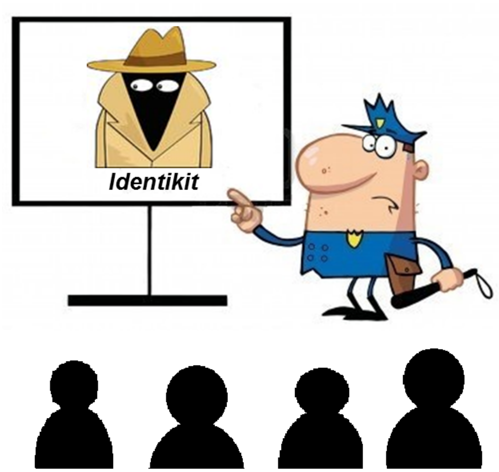
# How ? The tools

- Homomorphic encryption
- Blinding / obfuscation
- Oblivious transfer
- Garbled circuits
- Hybrid approach

- **Before that: a note on security definition**

# Security definition

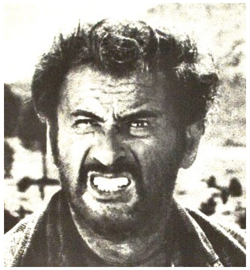
- What does security mean ?
- How do we prove security ?
- A huge zoo of security definitions exist
    - what do we want to impede to the attacker ?
    - what is the attacker allowed to know ?
    - what is the (computing) power of the attacker ?



# Security definition

- In a s.p.e.d. setting further details must be specified: will the adversary follow the protocol or not ?

Semi-honest (honest but curious) adversary: he follows the protocol but tries to infer secret information



Malicious (active) adversary: any action is allowed even departing from the protocol

Covert adversary: he is willing to deviate from the the protocol but does not want to be caught

# s.p.e.d. tools

- **Homomorphic encryption**
- **Blinding / obfuscation**
- Oblivious transfer
- Garbled circuits
- Hybrid approach

# The *homomorphic* paradigm

An algebraic operation on the plain messages is mapped into a (possibly different) algebraic operation on the encrypted messages

$$a \bullet b = D_{pr}[E_{pub}(a) \circ E_{pub}(b)]$$

$$if \begin{cases} \bullet = + \\ \circ = \times \end{cases} \Rightarrow a + b = D_{pr}[E_{pub}(a) \times E_{pub}(b)] \quad additive\ HE$$

$$Ka = D_{pr}[\underbrace{E_{pub}(a) \times E_{pub}(a) \ldots E_{pub}(a)}_{K\ times}] = D_{pr}[E_{pub}(a)^K]$$

# The *homomorphic* paradigm

With additive HE a number of interesting operators can be applied to signals:

$$\text{Component-wise encryption} \Rightarrow E[(a_1, a_2 \ldots\ a_n)] = (E[a_1], E[a_1] \ldots E[a_n])$$

$$\text{Scalar product (known vector)}: \ \langle \mathbf{a}, \mathbf{b} \rangle = \sum_{i=1}^{n} a_i b_i \Rightarrow E[\langle \mathbf{a}, \mathbf{b} \rangle] = \prod_{i=1}^{n} E[a_i]^{b_i}$$

$$\text{FIR filtering}: \ a_n = \sum_{k=1}^{L} a_{n-k} h_k \Rightarrow E[a_n] = \prod_{k=1}^{L} E[a_{n-k}]^{h_k}$$

$$\text{Linear transforms}: \ X_k = \sum_{i=1}^{n} a_{k,i} x_i \Rightarrow E[X_k] = \prod_{i=1}^{L} E[x_i]^{a_{k,i}}$$

# Peculiarities of HE

- Based on (probabilistic) public-key cryptography
  - Long keys
  - Samplewise encryption: large expansion factor
  - Complex operations with very large numbers
  - Ex. Pailler cryptosystem
    - key-length = 1024 bits (at least)
    - Cipher message = 2048 bits (at least)
    - Expansion factor = 8 for images, 2048 for bits
- No interaction for linear operations

# Non-linear (polynomial) functions and full HE

$$if\ \otimes and \oplus \exists : \begin{cases} a+b = D[E(a) \oplus E(b)] \\ a \times b = D[E(a) \otimes E(b)] \end{cases} \quad full\ HE$$

Kind of holy Graal in cryptography
recent breakthrough by Gentry

…

still impractical but rapidly improving

# Non-linear functions through blinding

- Example: how to square an encrypted number

$$E[x]$$

$$E[y] = E[x+b] = E[x]E[b]$$

$$\xleftarrow{E[y]}$$

$$y = D[E[y]]$$

$$g(y) = y^2 = x^2 + b^2 + 2xb$$

$$E[g(y)]$$

$$\xrightarrow{E[g(y)]}$$

$$E[x^2] = E[g(y) - b^2 - 2bx]$$

$$= E[g(y)]E[-b^2]E[x]^{-2b}$$

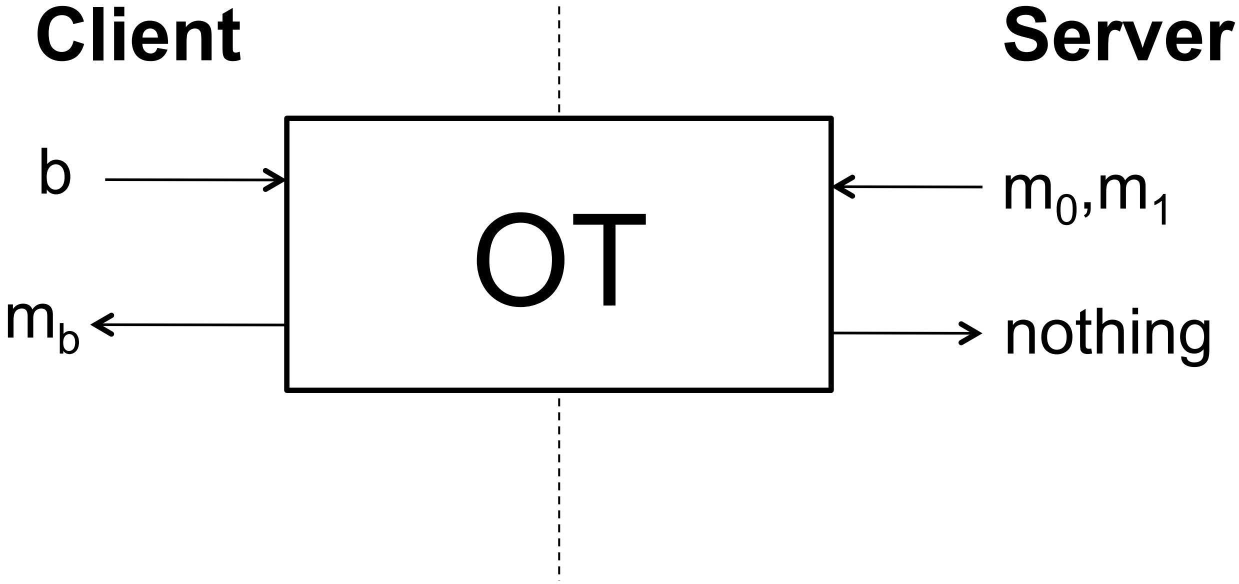
# SPED tools

- Homomorphic encryption
- Blinding / obfuscation
- **Oblivious transfer**
- **Garbled circuits**
- Hybrid approach

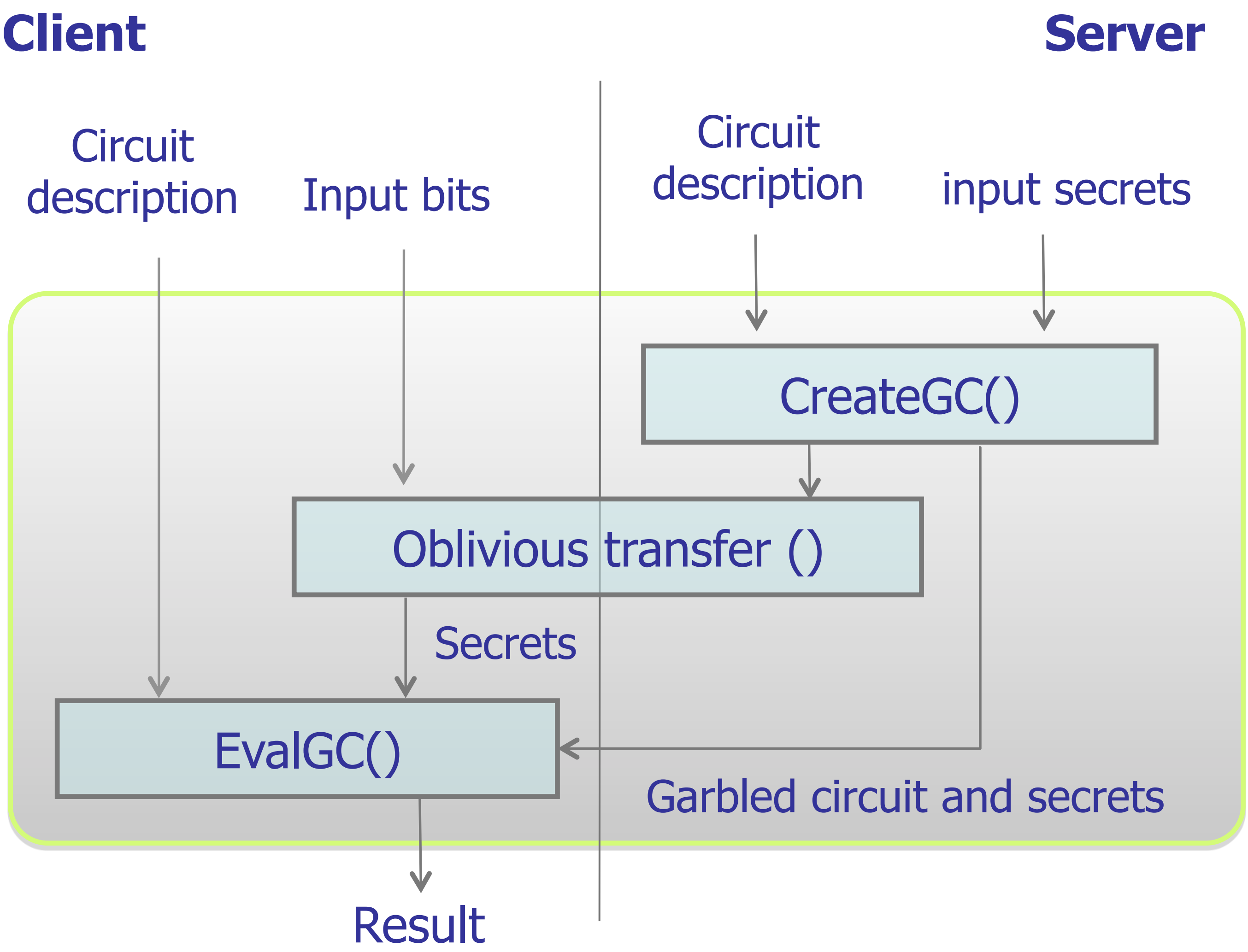
# An alternative approach: OT + GC

- Private computation of any function expressed as a Boolean (non recursive) circuit
- Symmetric cryptography
- Inputs at the bit level
- Thought to be impractical until 4-5 years ago
    - now: more than 100.000 non-free gates per second
        - Evans, D., et al. "Efficient privacy-preserving biometric identification."*Proceedings of the 17th conference Network and Distributed System Security Symposium, NDSS*. 2011.

# Oblivious transfer (OT)



# General structure of a GC protocol



# Peculiarities of GCs

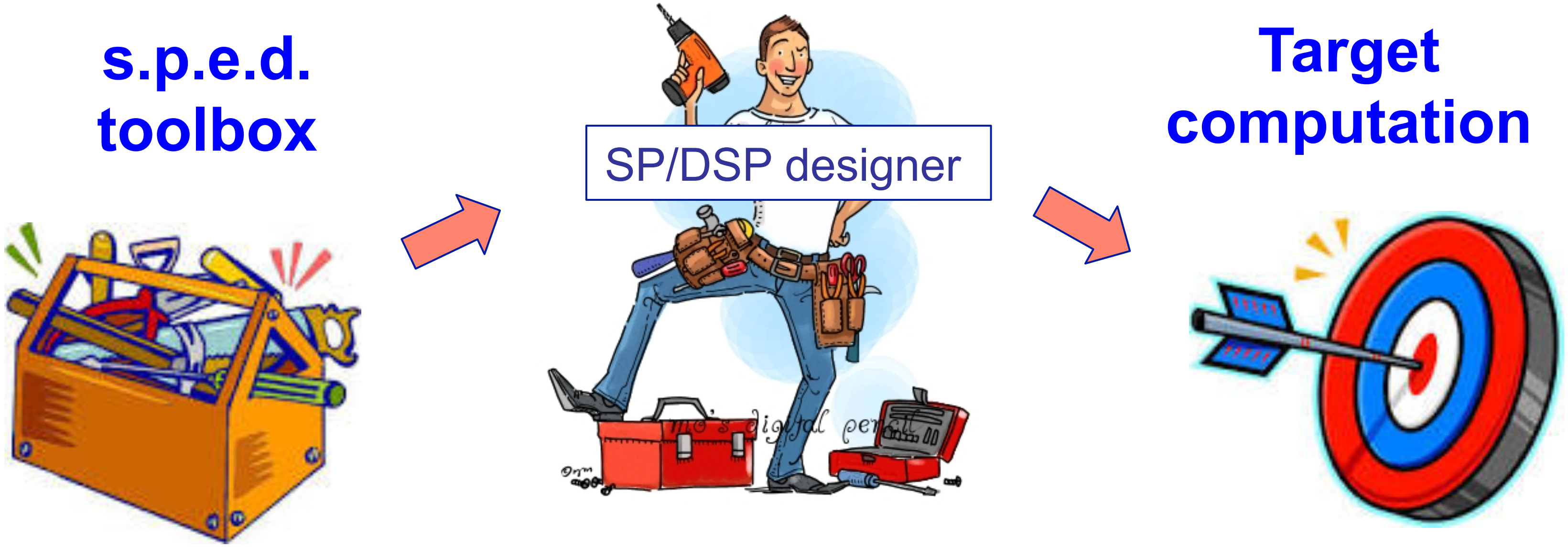
- Based on symmetric encryption -> light computation
  - Secrets = 80 bit long
  - Computation = hash functions
  - Offline computation
- Complexity grows with size of Boolean circuit
  - XOR gates come for free
  - Communication complexity maybe a problem

# Hybrid solution

- Most recent trend: hybrid solution
  - combine GC and HE
  - transcoding overhead

# What is left for SP designers: a lot

- The basic question IS NOT **if** a given functionality can be computed in a s.p.e.d. setting
- The basic question IS **how efficiently** a functionality can be evaluated in a s.p.e.d. setting (**computational, communication** and **round complexity**)



**s.p.e.d. toolbox**

SP/DSP designer

**Target computation**

# What is left for SP designers: a lot

- **Optimize algorithms**
  - Representation accuracy and number of variables
    - All cryptographic primitives work only on integer values -> data quantization necessary
    - Integer representation allowed but no truncation
    - Representation complexity may grow during the computation
  - Representation accuracy has a strong impact on
    - Accuracy of results
    - Complexity of the protocol
    - Trade-off needed

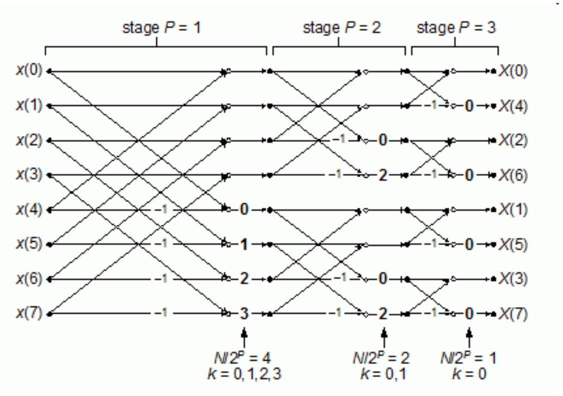
# Example: DFT vs FFT

- Integer representation
- Truncation is not possible
- Number of bits increases after each operation

- **DFT: many non-cascaded multiplications**

$$X(k) = \sum_{n=1}^{N} x(n)e^{j2\pi nk/N}$$

- **FFT less, cascaded, multiplications**



- In some cases DFT may be faster than FFT

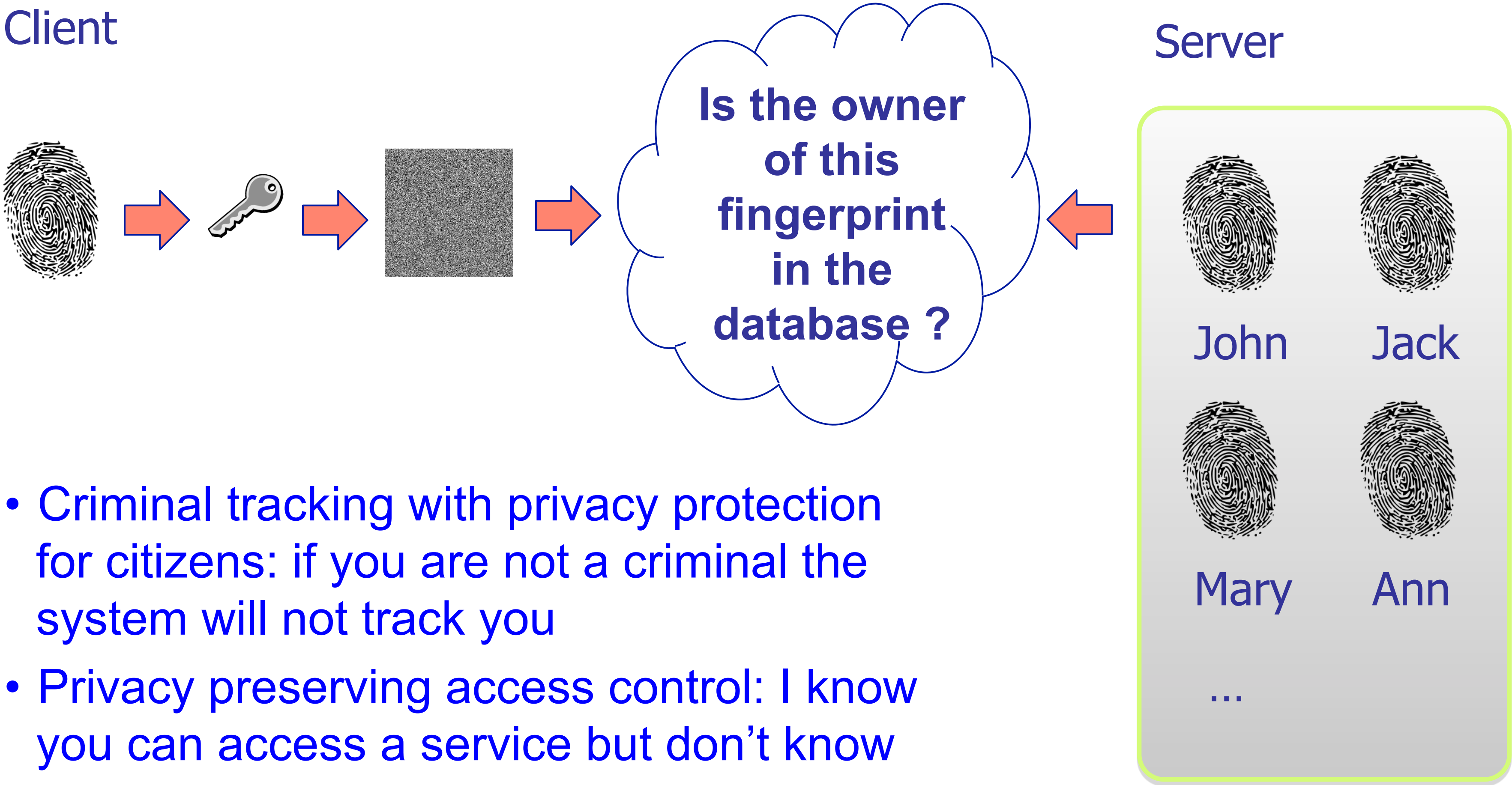
# What is left for SP designers: a lot

- **Optimize algorithms**
  - Basic operations used within the algorithms
  - Simple operations in the plain domain may be very complex when applied on encrypted signals
    - Comparisons, if-then-else, sorting: very complex operations with HE
    - Multiplications and divisions: very complex with GC

**THE - *RELATIVE* - PRICE TO PAY**
**TO PASS FROM PLAIN DOMAIN COMPUTATION**
**TO S.P.E.D. IS ALWAYS QUITE LARGE**

# A complete example
## (out of many possible ones)

# Biometric-based authentication



Client

Is the owner of this fingerprint in the database ?

Server

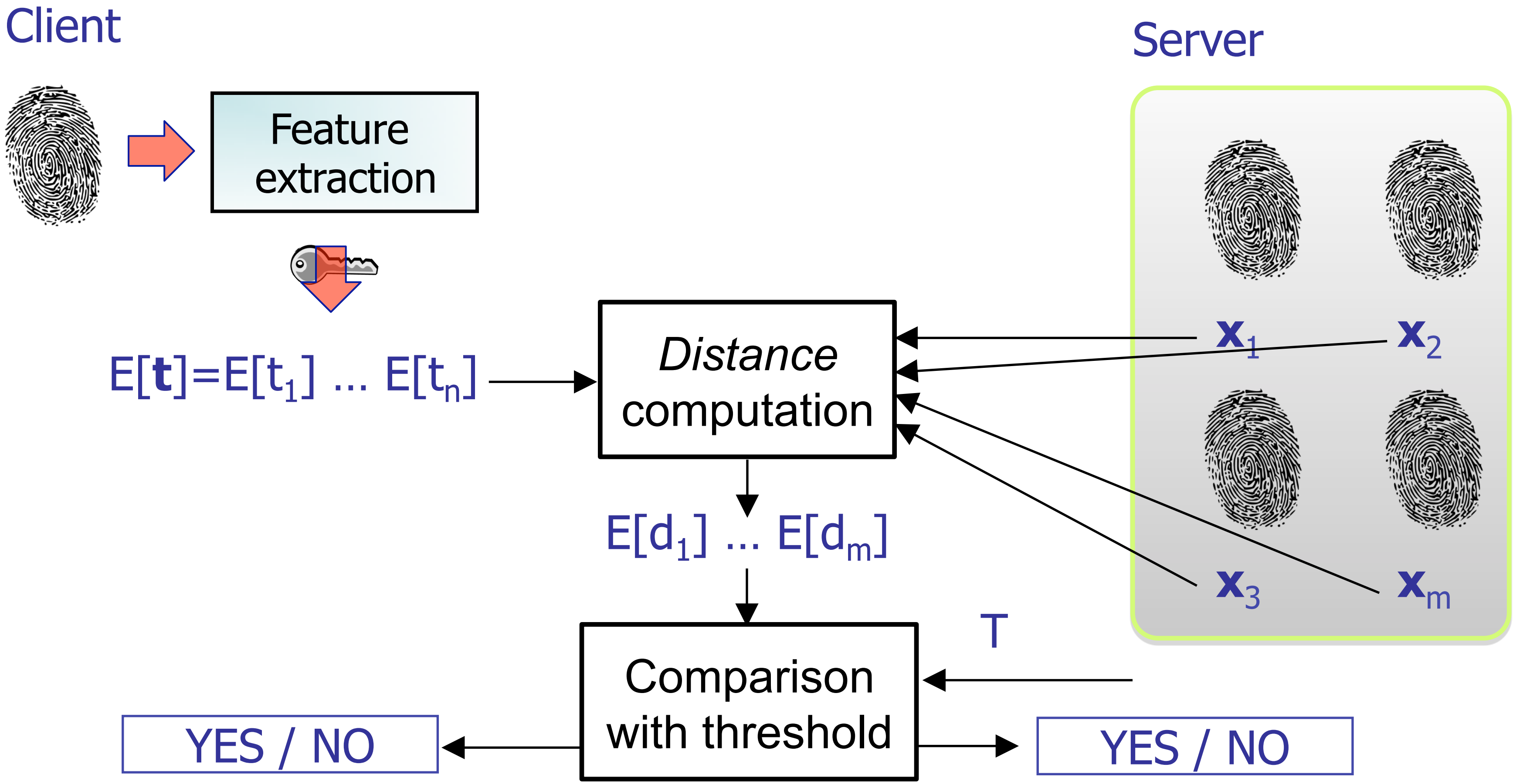
John Jack

Mary Ann

…

- Criminal tracking with privacy protection for citizens: if you are not a criminal the system will not track you
- Privacy preserving access control: I know you can access a service but don’t know who you are

# Biometric-based authentication

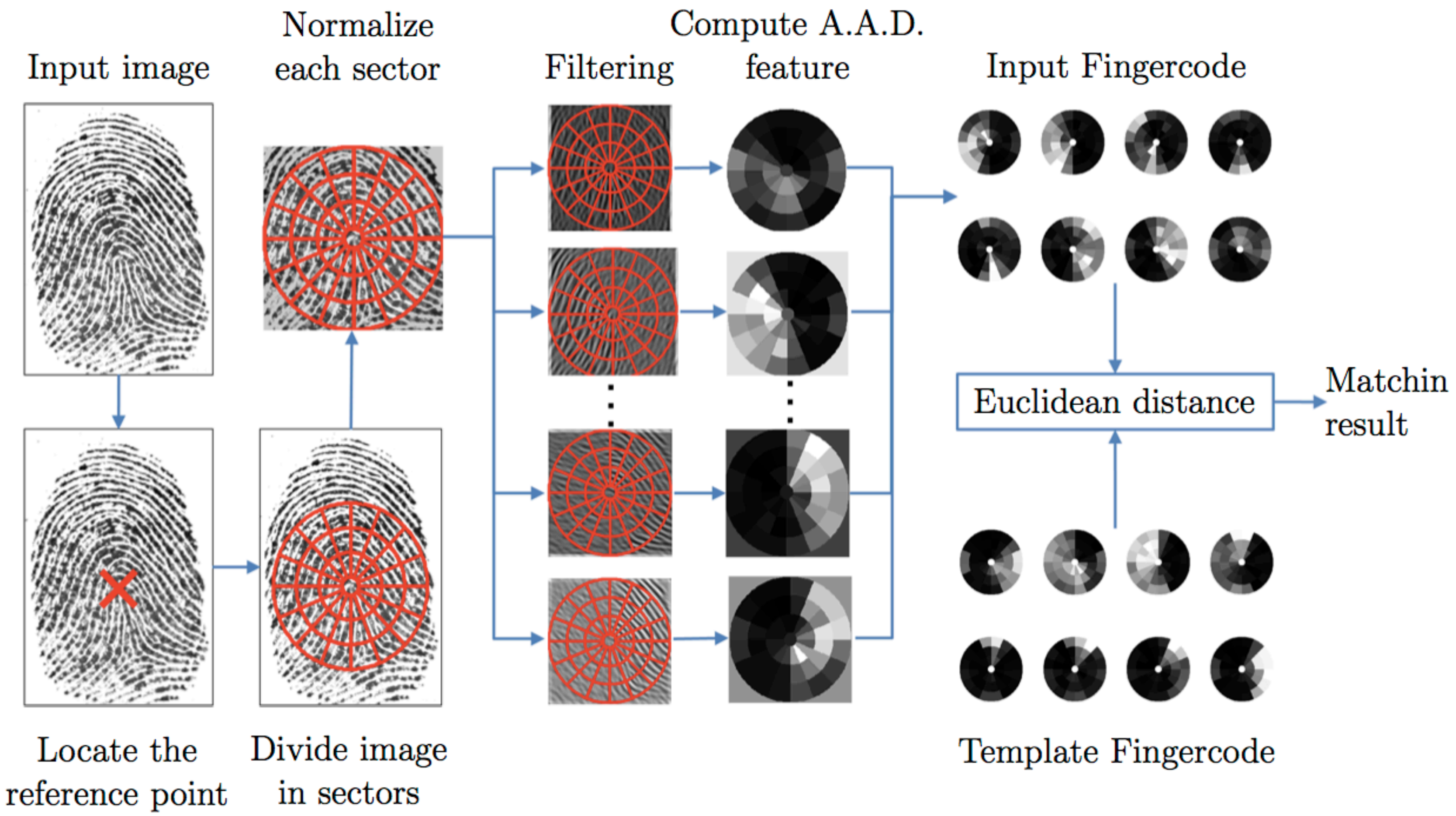


# SP choices

- Choice of feature set and distance function that ease an s.p.e.d. implementation
- Classical approaches based on minutiae not possible
- Our choice:
  - **Fingercode***
  - **Squared euclidean distance**

* M. Barni et al. “Privacy-preserving fingercode authentication” *Proceedings of the 12th ACM workshop on Multimedia and security*, 231-240, Rome, 2010.

# Fingercode representaion of fingerprints



# Optimization of fingercode representation

**Size of feature vector**

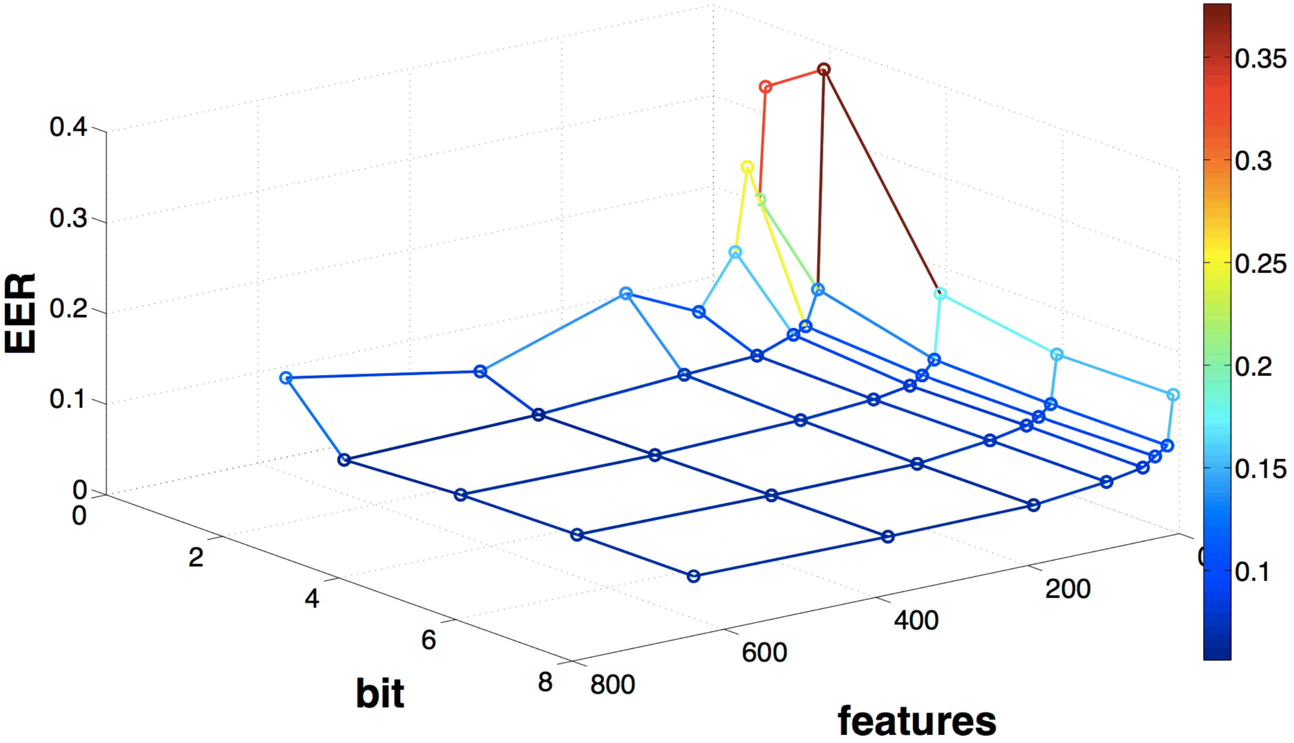
- $N_R$ = number of rings
- $N_A$ = number of arcs
- $N_S = N_R \times N_A$ = number of sectors
- $N_F$ = number of filters
- $N_V = N_F \times N_S$ = size of feature vector
- $N_\theta$ = number of rotated templates for enrolled user (9)

**Representation accuracy**

- $N_b$= number of bits for each feature (from 1 to 8)

# Optimization of fingercode representation

We evaluated the impact on matching accuracy (EER) by relying on a database with 408 fingerprints acquired by a CrossMatch verifier 300 sensor (500 dpi, 512x480 pixels).



# Selected configuration

**Size of feature vector**

- $N_R = 3$
- $N_A = 8$
- $N_S = 24$
- $N_F = 8$ (configuration C) or 4 (configuration D)
- $N_V = 192$ (C) or 96 (D)
- $N_\theta$ = number of rotate templates for enrolled user (9)

**Representation accuracy**

- $N_b$= 1bit, 2 bits

# Distance computation: classical approach

- The Squared Euclidean distance between an encrypted and a known vector is easy to compute by relying on HE

$$d(t,x)^2 = \sum_{1=1}^{n} (t_i - x_i)^2 = \underbrace{\sum_{i=1}^{n} t_1^2}_{\text{computed by the client}} + \underbrace{\sum_{i=1}^{n} x_1^2}_{\text{computed by the server}} - 2\sum_{i=1}^{n} \underbrace{t_i x_i}_{\text{computed by the server via HE}}$$

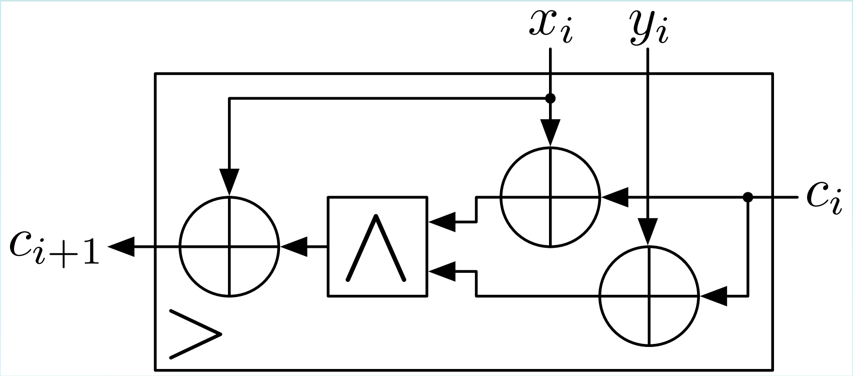
$$E[d^2] = E\left[\sum_{i=1}^{n} t_1^2\right] E\left[\sum_{i=1}^{n} x_1^2\right] \prod_{i=1}^{n} E[x_i]^{-2t_i}$$

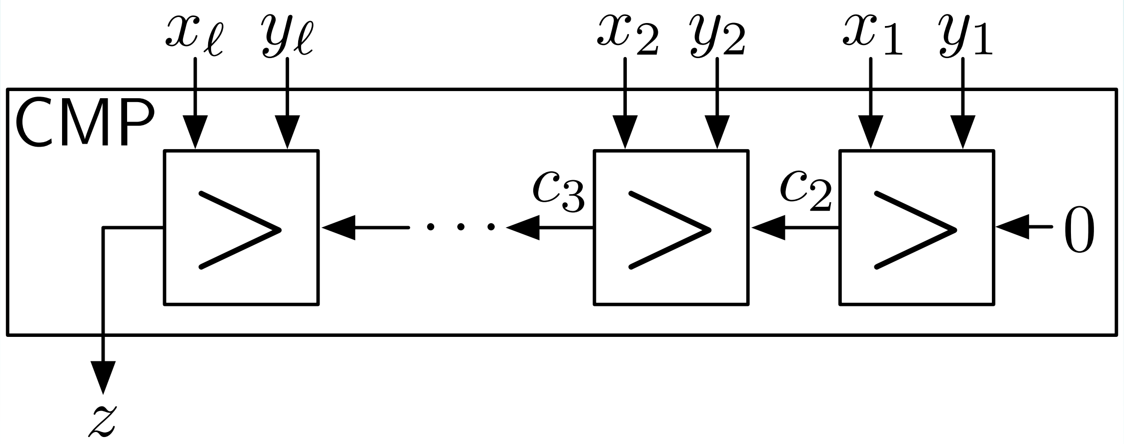
# Threshold comparison

- Comparison is by far easier through GC's

- Hybrid solution
  - distances computed via HE are converted into (secret) bits
  - Pass from HE to GC representation
  - Run the GC

# Clever circuit design





# Performance

- Set-up
  - Java-based implementation
  - PC-platform (clock 2GHz, RAM 2GByte)
  - Pailler + GC
  - 96 features, 4 bits per feature
- Complexity:
  - time: < 0.1 sec for template
  - bandwidth: 100Kbit per template
- Similar performance with
  - face recognition, iris recognition

# Conclusions: a roadmap for future research

- Efficiency, efficiency, efficiency
  - Crypto-level: more efficient primitives
  - **SP level**
    - **s.p.e.d. oriented algorithm design**
    - **Ad-hoc security measures**
- Security against malicious adversaries
  - recent breakthrough: GC construction against malicious adversary at 11500 gates/s
    - Nielsen, Jesper Buus, et al. "A new approach to practical active-secure two-party computation." *Advances in Cryptology–CRYPTO 2012*. Springer Berlin Heidelberg, 2012. 681-700.
- **System-level solutions, new applications**
- **Multi-disciplinary training, awareness raising**

# Thank you
# for your attention