Security and Authenticity for the Medical Document Image

Daniel HO
International Planning Division
Nikon Hong Kong Limited

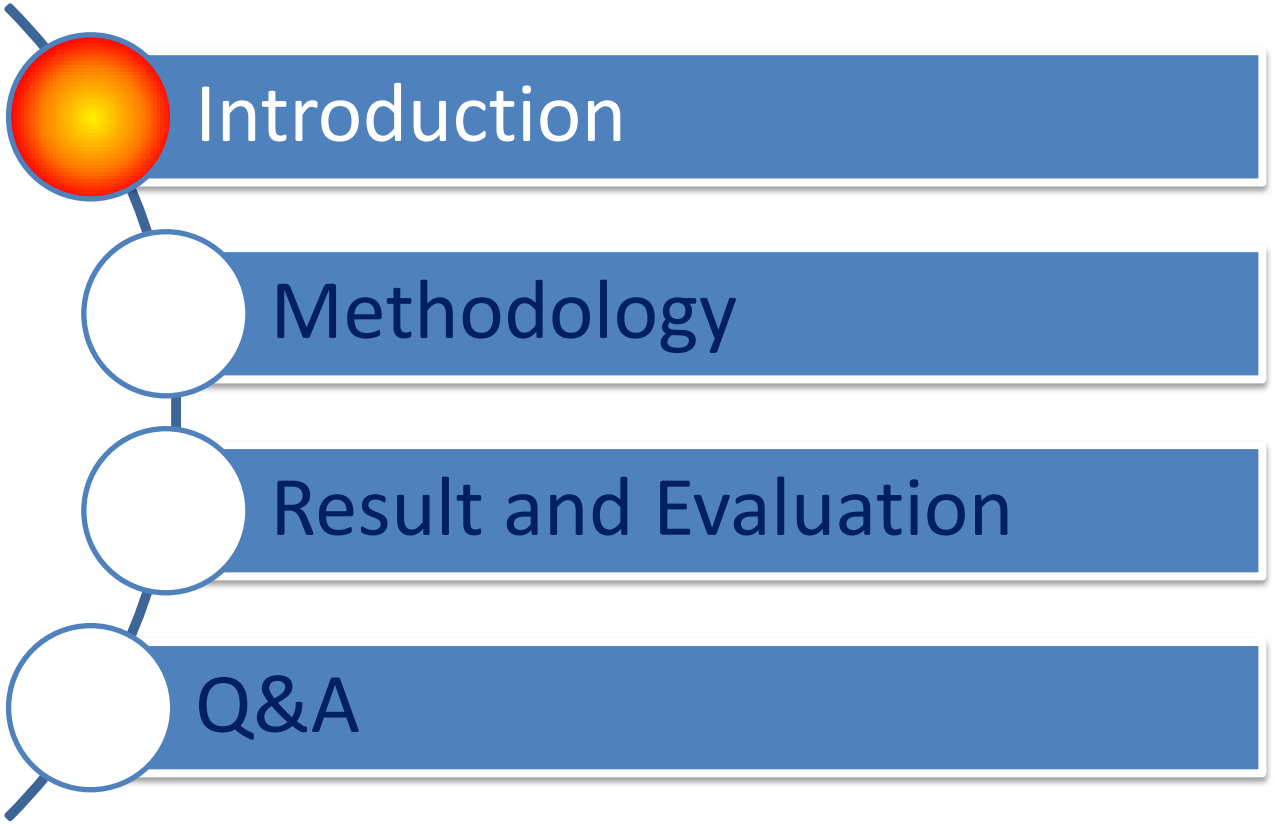
Peter LO
Consultant
Fresenius Netcare

Prof. Qin LU
Professor & Associate Head
Department of Computing
The Hong Kong Polytechnic University

Agenda

- Introduction
- Methodology
- Result and Evaluation
- Q&A

Agenda



Introduction

The Electronic Medical Record (EMR):

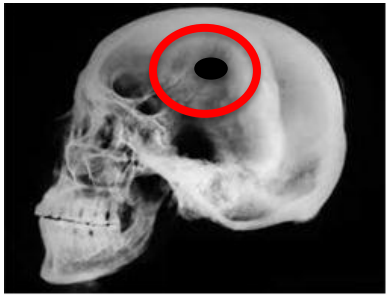
- **Used often in modern world;**
- **More flexible and readily available comparing to paper-based records;**
- **Image data is important evidence.**

However,

- **Images can be tampered easily with editing tools**

Therefore,

- **Proof of authenticity of the image becomes important;**

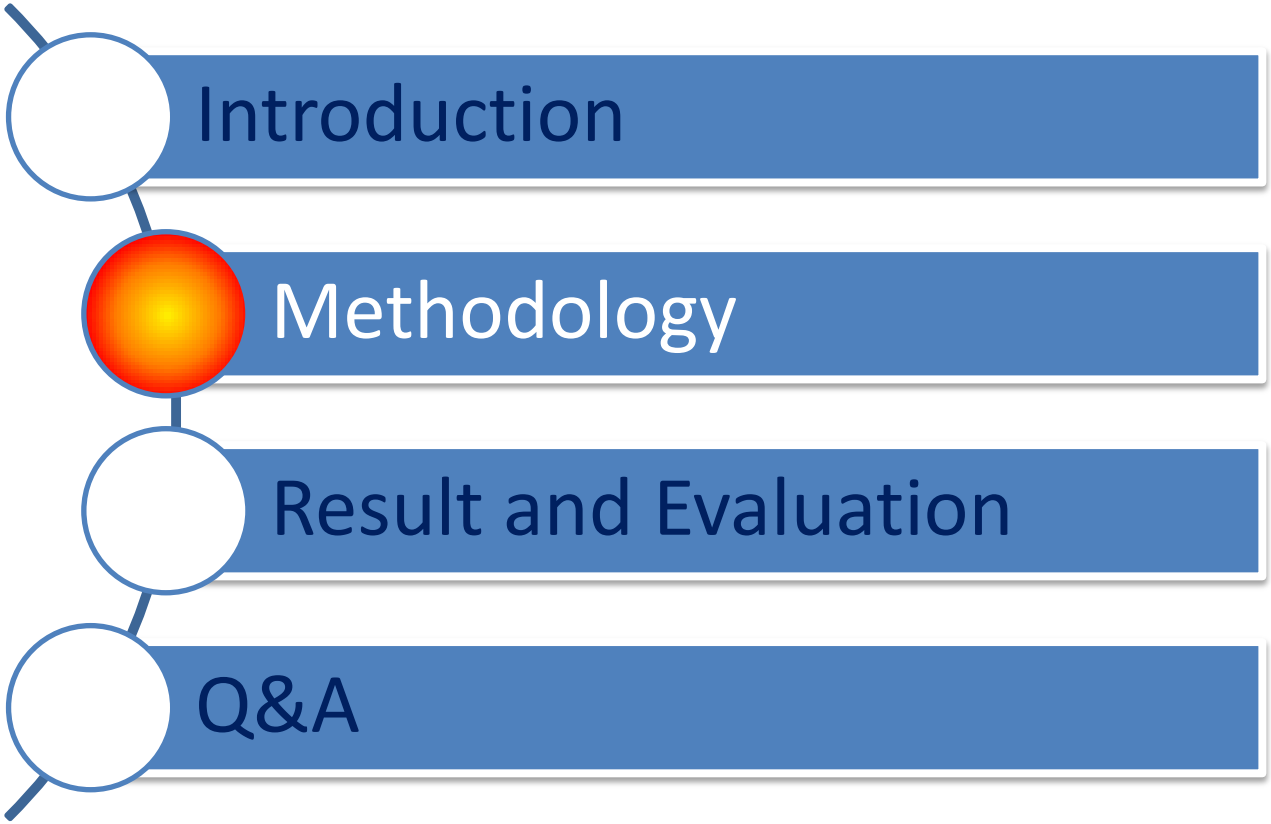


Potential Image Data

Resistance for Digitized Image

- Digitized images can be modified easily by software like Photoshop
- **Risks**: If an archive of a case history for a patient with result "**+**" can be modified to become "**-**", the consequence can be disastrous.

Agenda



Solution

- Digital Photo Modification Detector (DPMD)
  - Encrypt the digital signature by using the digital image ballistics from JPEG data itself so that alteration to image files in point to point transfer can be detected

Components of DPMD

- Two major components in DPMD
  - Digital Signature Generator (DSG)
  - Digital Image Detector (DID)

Digital Signature Generator (DSG)

- Once a digital image is created, the image should first be passed to the DSG and a digital key is generated using the EXIF metadata header and features unique to the JPEG image
- Then the digital key is encrypted and embedded into the original JPEG image

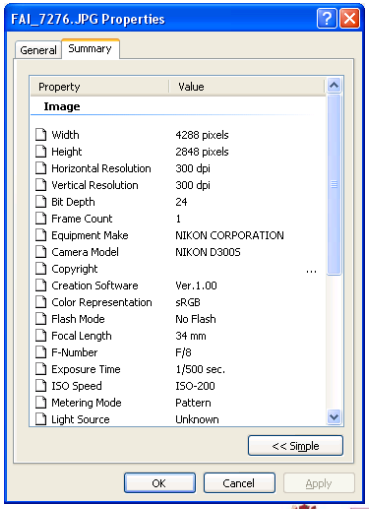
Workflow for DSG

1.) Scan image by digital instrument

2.) Generate Digital Signature

a.) Digital image in JPG

b.) EXIF of the image



3.) Create the Protected Image (DPMD-JPG file)

a.) Put the word "***DPMD***" into the EXIF-Tag ~ **'Creation Software'** & insert the "***Encrypted Digital Signature***" into the EXIF-Tag ~ '**User Comment**'.

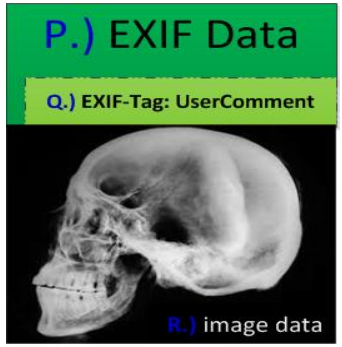
c.) Omits the field of "**Creation Software**" & **"User Comment"** during general the digital signature

d.) **Private Key** = "Equipment Maker" + "Camera Model" + "SerialNumber(Key-in temporary)"

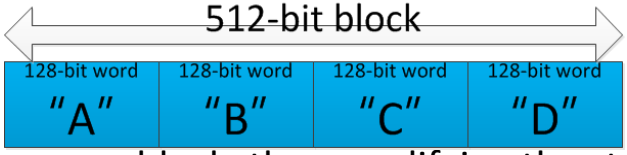
e.) Encrypt the "Digital" signature" by the **Private Key**

Step 1: Generate Digital Signature by MD5

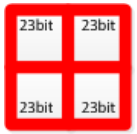
➢ Generate a 128bit fixed-length hash for any inputted file



1. Initial the EXIF-Tag: UserComment as “Blank” of the inputted file
   - P + R ➔
2. Breakdown the initialed image file into a crowd of 512-bit msg. blocks
3. The main algorithm operation based on 128-bit state as below:



4. Each turn process a block, then modifying the state
5. Each block process have 4 similar stages called “rounds”
6. Each “round” composed of 16 similar operations
   - based on a non-linear function “ F ”
   - modular addition, and left rotation. $\lll_s$
   - Figure 1 illustrates one operation within a round.



7. There are four possible functions F : (F, G, H, I)
   - a different one is used in each round:

$$F(X,Y,Z) = (X \wedge Y) \vee (\neg X \wedge Z)$$
$$G(X,Y,Z) = (X \wedge Z) \vee (Y \wedge \neg Z)$$
$$H(X,Y,Z) = X \oplus Y \oplus Z$$
$$I(X,Y,Z) = Y \oplus (X \vee \neg Z)$$

$\oplus, \wedge, \vee, \neg$ denote the XOR, AND, OR and NOT operations respectively.

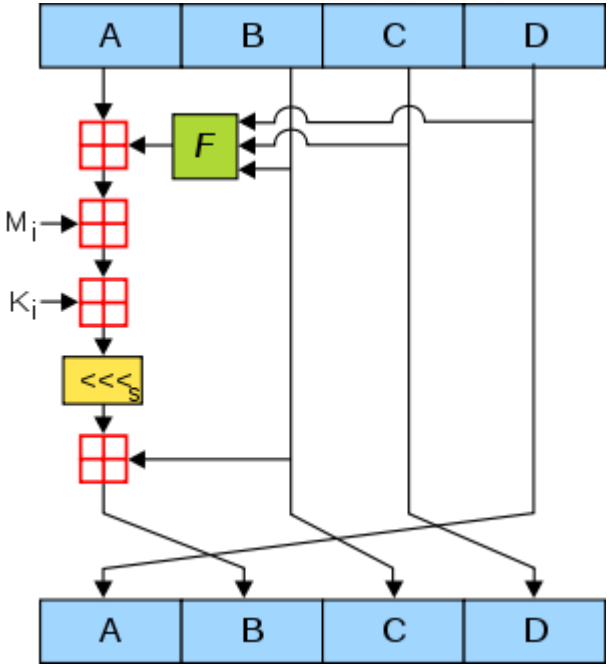


Figure 1:

Ref.: http://osix.net/modules/article/?id=507
http://en.wikipedia.org/wiki/MD5

Step 1: Generate Digital Signature by MD5 (Cont')

Source image file	Outputted Digital Signature
	9e107d9d372 bb6826bd81d 3542a419d6
	e4d909c290d 0fb1ca068ffad df22cbd0

Different Digital Signatures will be generated if data is modified !

Step 2: Digital Signature Cryptography with RSA

- **Encrypt & Decrypt any Digital Signature from the inputted image**
- **RSA 4096bit scheme**

Encryption:

- **Public key (*n*, *e*)** can be published; **private key (d)** must keep in secret.
- Let **M** be the original msg. **C** be the Encrypted msg.
  - Turns **M** into an integer **m**, such that **0 < *m* < *n*** by using padding scheme.
  - Then computes the ciphertext ***c*** :

$$c = m^e \pmod{n}.$$

Decryption:

- Original msg. ***m*** can be recovered from ***c*** by using the private key exponent ***d*** as below.

$$m = c^d \pmod{n}.$$

- Given ***m***, the original message **M** can be recovered by reversing the padding scheme.

Ref.: http://en.wikipedia.org/wiki/RSA
http://mathworld.wolfram.com/RSAEncryption.html

Step 2: Digital Signature Cryptography with RSA (Cont')

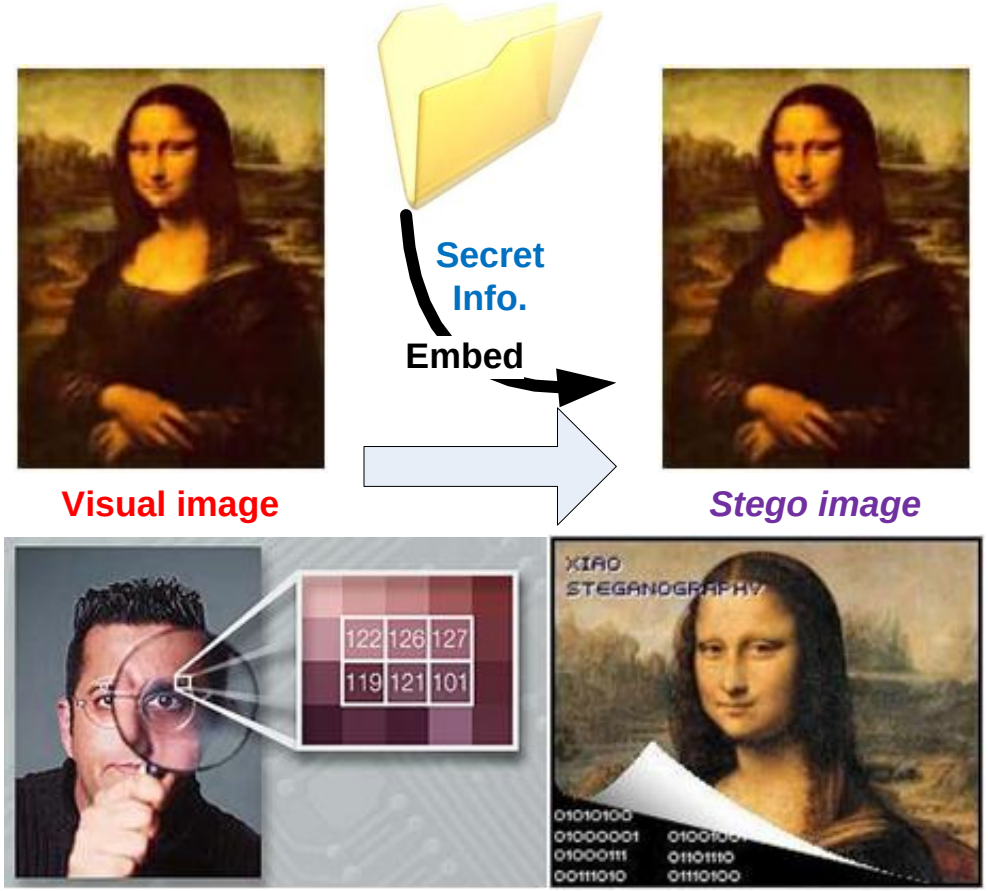
Key generation:

- Choose 2 different prime numbers ***p*** and ***q***.
  - ***n*** ← the modulus of the public and private keys
- Compute $n = pq$.
- Compute $\phi(n) = (p-1)(q-1)$, ← ϕ is Euler's totient function.
- Choose an integer ***e***,
  - $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1$, ← ***e*** and $\phi(n)$ are coprime.
  - ***e*** is released as the public key exponent.
  - ***e*** → short bit-length → small Hamming weight results → more efficient encryption
  - But, if the values of ***e*** is small, (such as 3) ➔ less secure .
- Determine $d = e^{-1} \bmod \phi(n)$; i.e. ***d*** is the multiplicative inverse of $e \bmod \phi(n)$.
  - This is more clearly stated as solve for **d** given $(d*e) \bmod \phi(n) = 1$
  - This is often computed using the extended Euclidean algorithm.
  - ***d*** is kept as the **private key** exponent.
- The **public key** consists of the modulus ***n*** and the public (or encryption) exponent ***e***. The **private key** consists of the private (or decryption) exponent ***d*** which must be kept secret.

Ref.: http://en.wikipedia.org/wiki/RSA
http://mathworld.wolfram.com/RSAEncryption.html

Step 3: Store the Digital Signature with Steganography

- **Steganography**
- Hide the encrypted signature inside EXIF
- EXIF-Tag
  - UserComment
  - CreationSoftware (*Program Name*)
- Stego image looks same as original image



Step 3: Store the Digital Signature with Steganography (cont')

[FileName.jpg]

To Generate "DPMD-JPG":

1. Input a source image
2. initial **Q** into Blank
3. ***Digital Signature*** **= MD5(P + R)**
4. **Q = encrypt(*Digital Signature*)**
5. Store **Q** back to the EXIF-Tag: **UserComment**
6. *Stego image file has been created!*

➔ This **Stego image** known as **"DPMD-JPG"**

Digital Image Detector (DID)

- When the image is transmitted to the target point, DID will validate the digital image against the embedded digital key
- If the image or the EXIF data is altered, the digital key which is unique to the original image would not find the match in the decoding process, thus alteration can be detected

Workflow for DID

1.) Check the inputted jpg Photo is DPMD or not

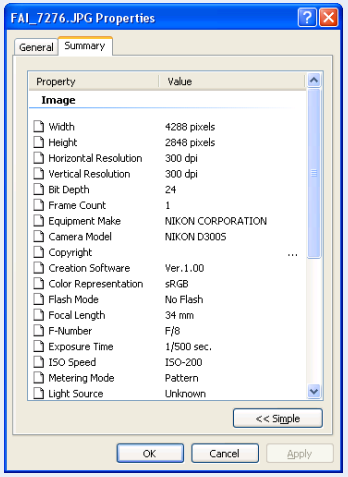
a.) Check are the Photo created by "**DSG**" or not? By checking the word "**DPMD**" existing in the Field of "**Creation Software**" under EXIF or not.

2.) Check the jpg Photo is DPMD or not. Generate the Digital Signature if it is

3.) Compare two Digital Signature. (AttachedOne vs. GenerateOne)

4.) Output the Result

a.) EXIF of the image

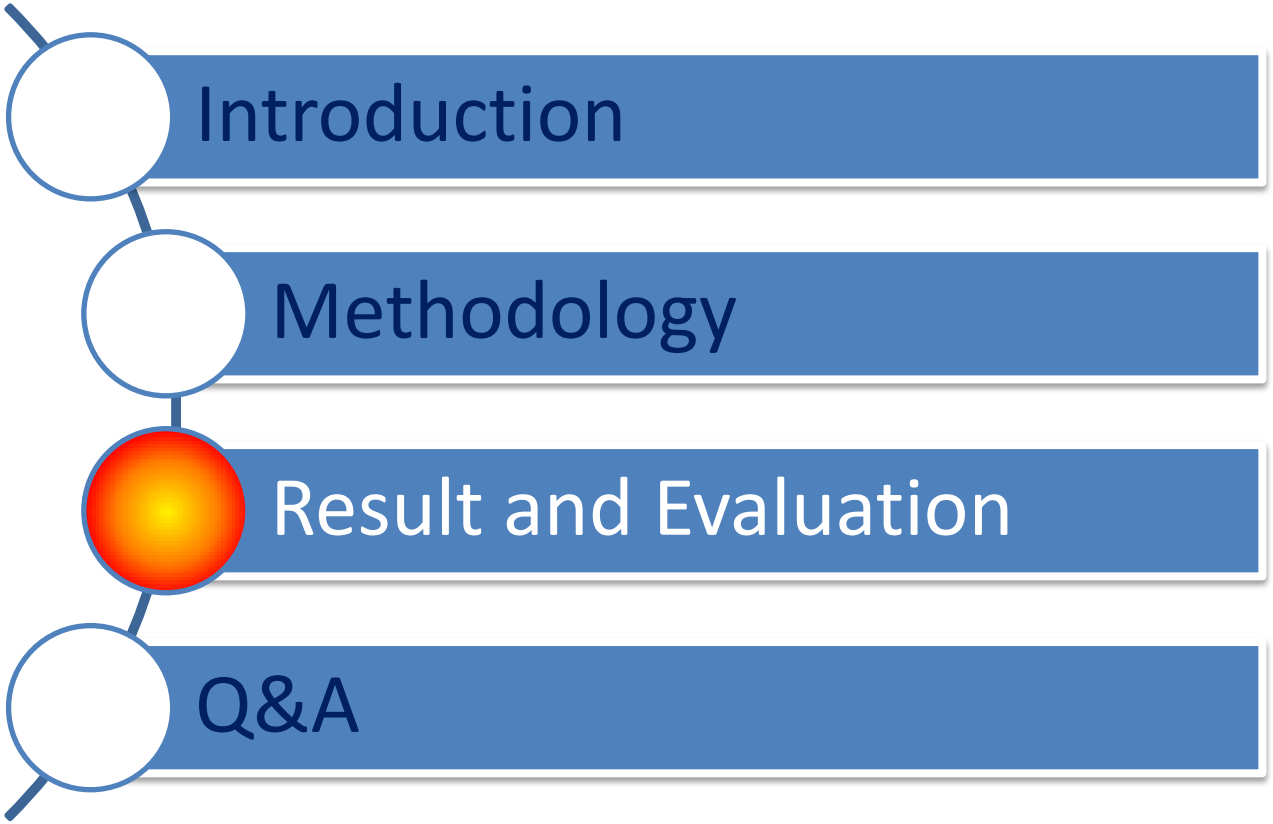


b.) Omits the field of "**Creation Software**" during general the digital signature

c.) **Key** = "Equipment Make" + "Camera Model" + "SerialNumber(Key-in temporary)"

d.) Encrypt the "Digital" signature" by the **Key**

Agenda



Testing Example

Original image

Original image

Verification by DPMD

Verification result of DPMD

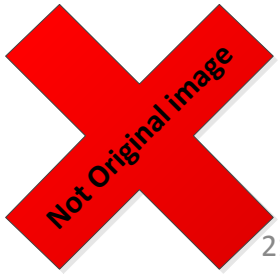
Original image

image modification

Modified image

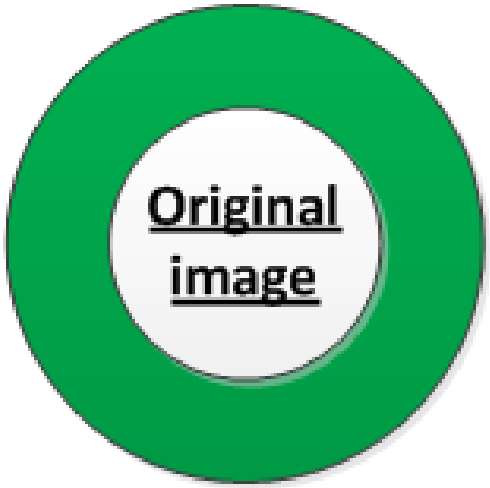
Verification by DPMD

Verification result of DPMD



Evaluation

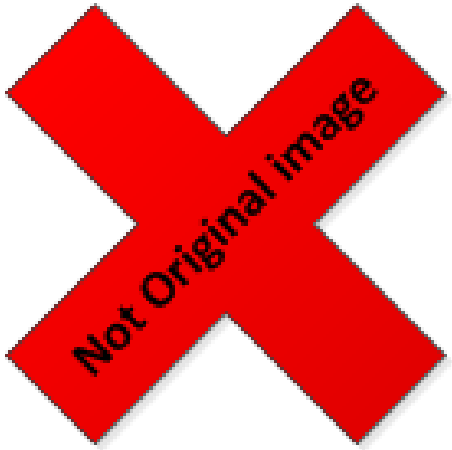
Instance will be classified as original image by DPMD:



- **Original image**
  - Original image;
  - Renamed image files;
  - All true copies of the original image files;
  - Renamed true copies

Evaluation

Instance will NOT be classified as original image by DPMD:



- **Modified image**
  - Modified image data;
  - Modified EXIF Data;
  - Re-save image by third parties' software;
  - Not DPMD format image file

Testing Scenario and Result

No.	Scenario	Result
1	No modification	Pass
2	EXIF (image properties) is modified	DPMD detect successfully
3	Histogram is modified	DPMD detect successfully
4	Image is modified	DPMD detect successfully
5	Resizing	DPMD detect successfully
6	Image is saved by other software (without edit)	DPMD detect successfully

Application

DPMD as an Apps

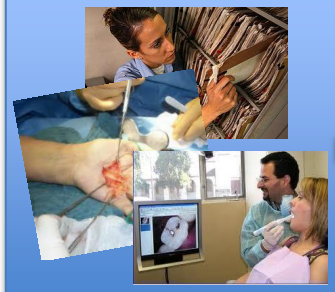

Medical Record


Image Digitization


Electric Medical Record


Protected with DPMD


Electric Medical Record
(with DPMD Protection)

DPMD embedded in Hardware

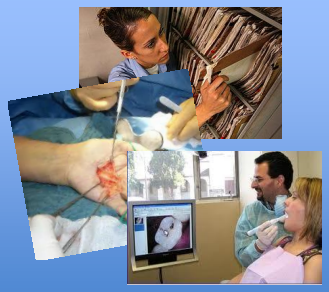

Medical Record

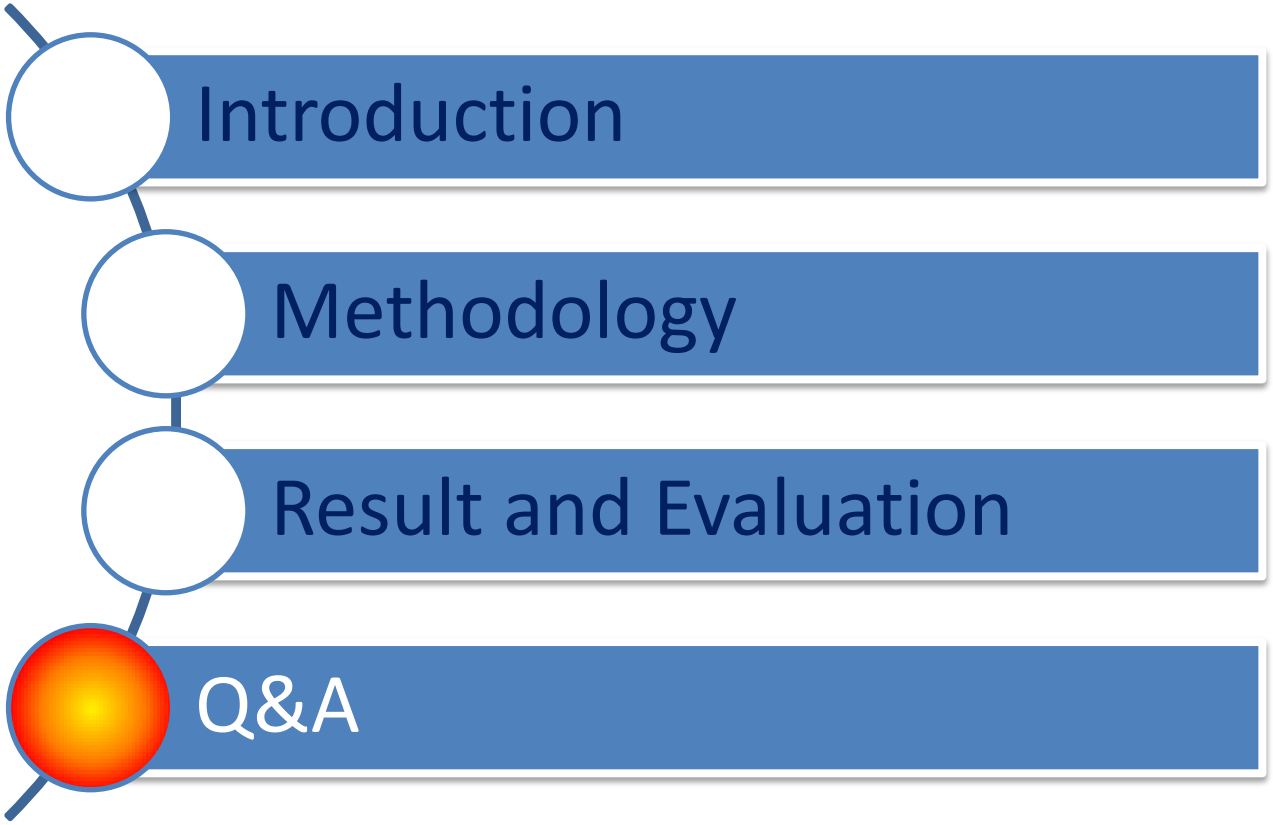

Image Digitization
(with embed DPMD)


Electric Medical Record
(with DPMD Protection)

Conclusion

- ✓ Modified image can be detected successfully by DPMP
- ✓ To prevent unauthorized image alteration, DPMD can be a tool for image modification detection
- ✓ A solution for increasing the credibility of digital image in EMR

Agenda




Thank you!