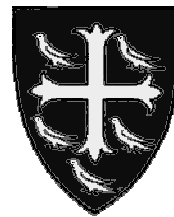


# Non-standard Authentication

Long Nguyen and Bill Roscoe

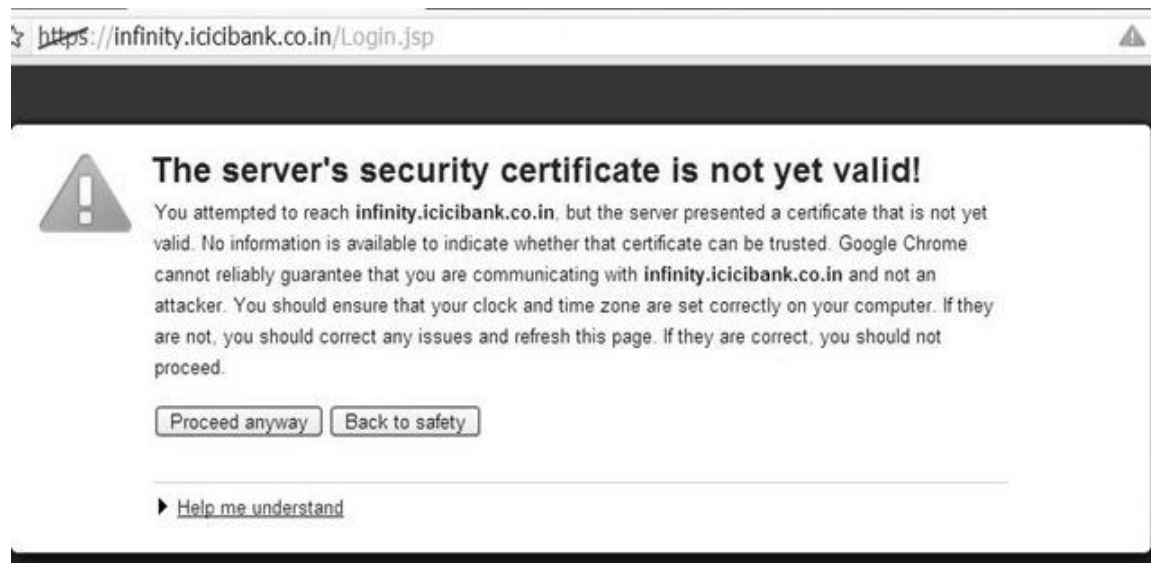
Oxford University Computing Laboratory  
University College



# Authentication

Authentication: proving that a communication is from the right device or person.

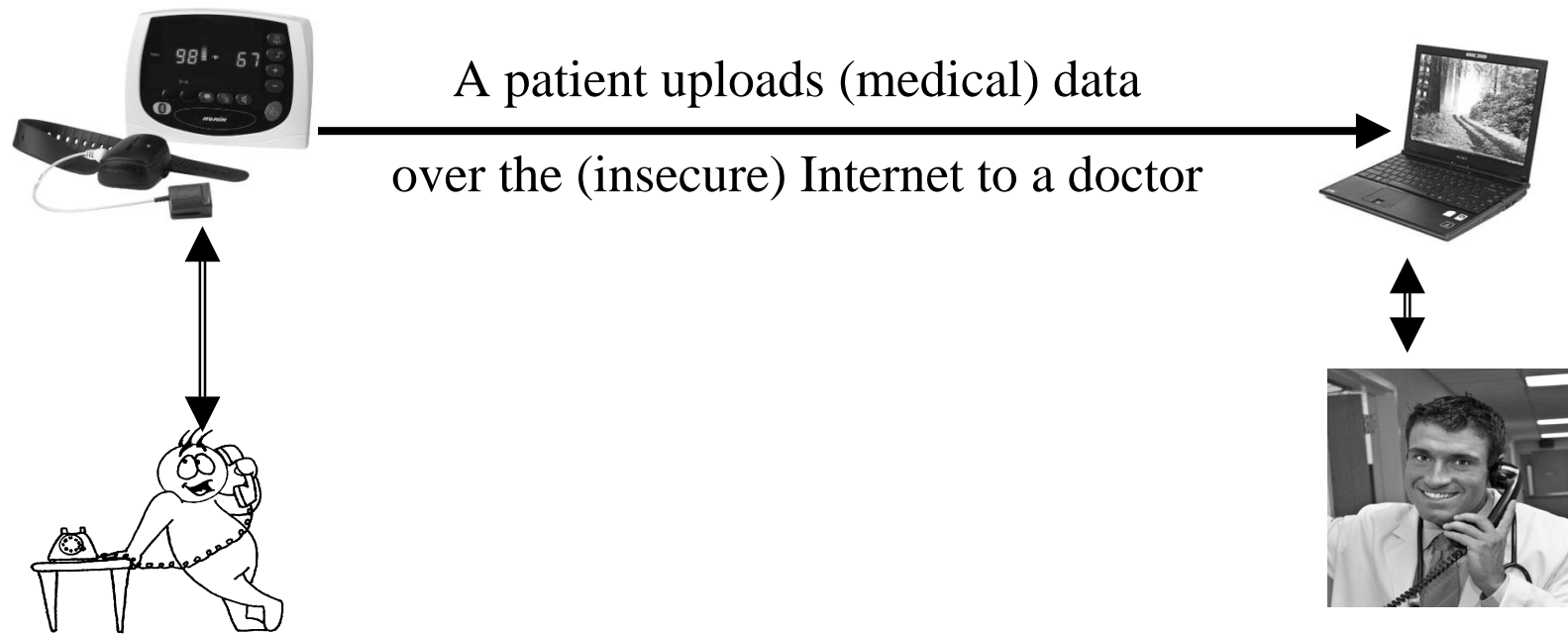
- In tradition: passwords (Bluetooth) and security certificates (PKI).
- Disadvantages: human misuse and misunderstanding.
- Low-power devices: RFID, smart card, mobile phone, medical sensor...



# New authentication technology

Building security on top of human trust

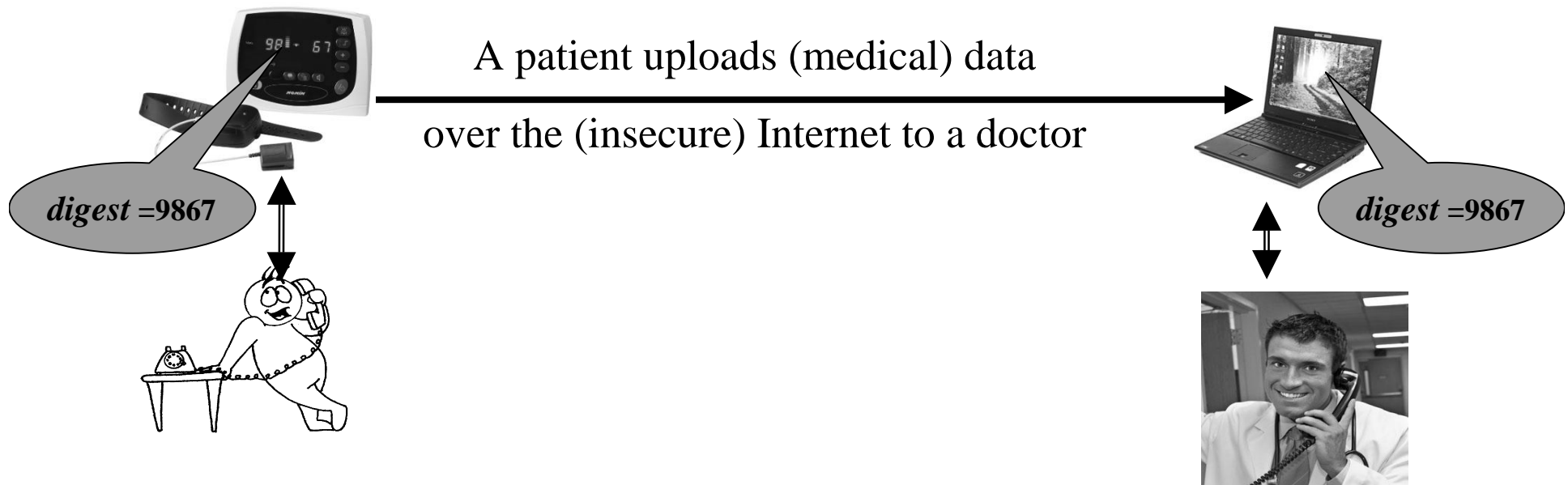
**Telephone-medicine:** a patient wants to upload medical data (e.g. medical images) from a medical sensor (a low-power device) to a doctor's laptop.



# New authentication technology

Building security on top of human trust

**Telephone-medicine:** a patient wants to upload medical data (e.g. medical images) from a medical sensor (a low-power device) to a doctor's laptop.

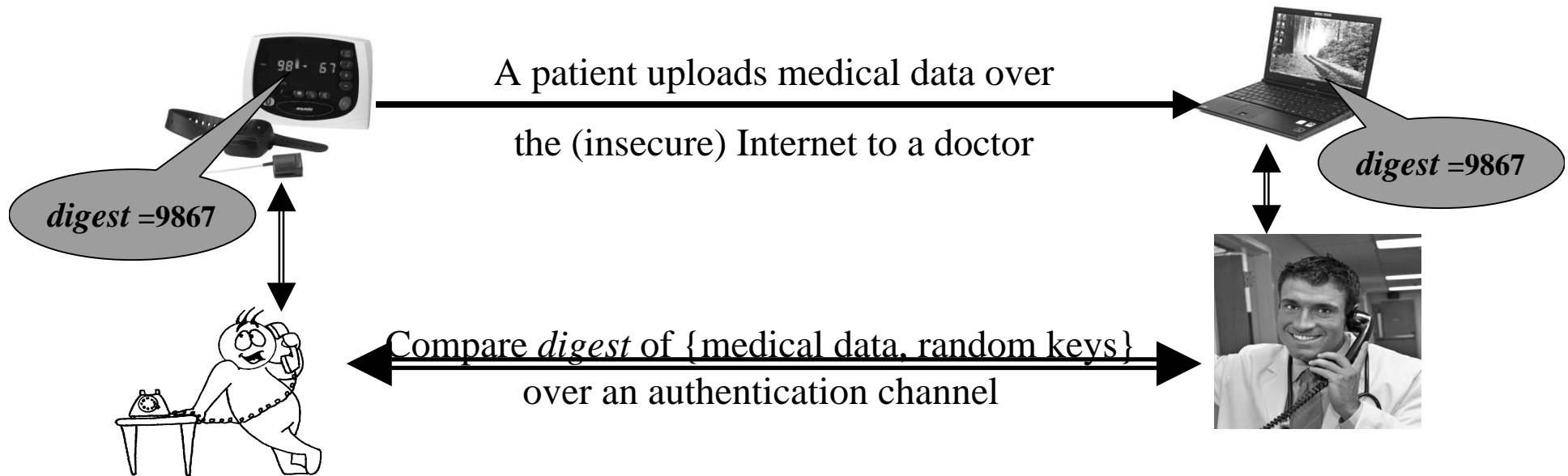


The short digest depends on: - medical data (INFO);

- random keys  $k$ 's introduced by the devices. <sup>4</sup>

# New authentication technology

An application in telephone-medicine

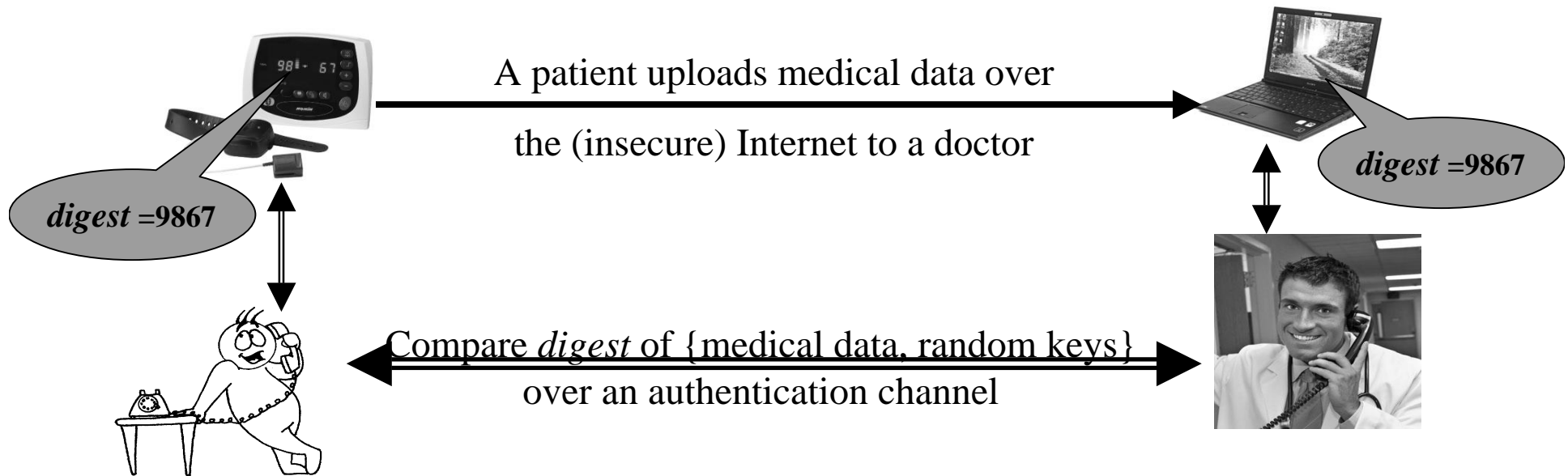


Human trust and digest equality over authentication channel imply:

- the doctor and his laptop are authenticated to the patient;
- integrity of (medical) data;
- no need for security certificates.

# New authentication technology

## An application in telephone-medicine



Human trust and digest equality over authentication channel imply:

- the doctor and his laptop are authenticated to the patient;
- integrity of (medical) data;
- no need for security certificates.

The challenges:

- digest needs to be short to reduce human interactions.
- digest must be *random & unpredictable* until it is displayed.

# Authentication channels

- Two kinds of channel used in the technology:
  - Dolev-Yao channel: insecure and high-bandwidth WiFi or the Internet
  - ⇒ authentication channel: authenticity, integrity, but not confidentiality
- Examples of authentication channels are telephone/human conversations, text messages, and manual data transfers.

# Authentication channels

- Two kinds of channel used in the technology:
  - Dolev-Yao channel: insecure and high-bandwidth WiFi or the Internet
  - ⇒ authentication channel: authenticity, integrity, but not confidentiality
- Examples of authentication channels are telephone/human conversations, text messages, and manual data transfers.
- Authentication channels are limited in bandwidth: we need to optimise the number of bits required to be sent over the authentication channels.
- If parties manually compare a string of  $b$  bits then we want the following:

$$\text{Prob}[\text{a successful one-shot attack}] \leq 2^{-b}$$

## *digest*: a new cryptographic function

- ( $b$ -bit output) *digest* functions have similarities to *universal hashes*.

As key  $k$  varies uniformly:

- for any fixed message  $m$ ,  $\text{digest}(k, m)$  is uniformly distributed.
  - for any fixed  $m_1 \neq m_2$ :  $\text{Prob}_k[\text{digest}(k, m_1) = \text{digest}(k, m_2)] \leq \varepsilon$
- *digest* has a short output (e.g.  $b = 16$  bits): will probably be (significantly) faster to compute than cryptographic hashes: 160-bit SHA, MD5.

# MANA I: one-way authentication

(Gehrman-Mitchell-Nyberg)

1.  $A \longrightarrow B: INFO$

1'.  $B \Rightarrow A: 1\text{-bit ACK}$

2.  $A \Rightarrow B: K, \text{digest}(K, INFO)$

- $A$  wants to authenticate data  $INFO$  to  $B$ .
- digest and key  $K$  are  $b$  bits, so  $A$  transmits  $2b$  bits over authentication channel.

## Security analysis of MANA I

1.  $A \longrightarrow I(B): INFO$

$I(A) \longrightarrow B : INFO'$

1'.  $B \Longrightarrow A : 1\text{-bit ACK}$

2.  $A \Longrightarrow B : K, \text{digest}(K, INFO)$

- Since digest is only  $b$  bits:

$$\text{Prob[ a successful collision attack]} \geq 2^{-b}$$

- With  $2b$  authentication bits, this is *sub-optimal* in human work.

## Security analysis of MANA I

1.  $A \longrightarrow I(B): INFO$

$I(A) \longrightarrow B : INFO'$

1'.  $B \Rightarrow A : 1\text{-bit ACK}$

2.  $A \Rightarrow B : K, \text{digest}(K, INFO)$

- Since digest is only  $b$  bits:

$$\text{Prob[ a successful collision attack]} \geq 2^{-b}$$

- With  $2b$  authentication bits, this is *sub-optimal* in human work.
- Since the  $b$ -bit key  $K$  is too short, digest collision probability  $\epsilon \gg 2^{-b}$

## Improved MANA I

1.  $A \rightarrow B$ : *INFO*, Hash( $k$ )

1'.  $B \Rightarrow A$ : 1-bit ACK

- A long 160-bit key  $k$  is used, but it can be transmitted over the Internet.

## Improved MANA I

1.  $A \longrightarrow B: INFO, \text{Hash}(k)$

1'.  $B \Rightarrow A: \text{1-bit ACK}$

2.  $A \Rightarrow B: \text{digest}(k, INFO)$

3.  $A \longrightarrow B: k$

- A long 160-bit key  $k$  is used, but it can be transmitted over the Internet.
- Order of Messages 2 and 3 is not important.
- A  $b$ -bit authentication string optimises human work:

$$\text{Prob}[\text{a successful collision attack}] \approx 2^{-b}$$

## Improved MANA I

1.  $A \longrightarrow B: INFO, \text{Hash}(k)$

1'.  $B \Rightarrow A: 1\text{-bit ACK}$

2.  $A \Rightarrow B: \text{digest}(k, INFO)$

3.  $A \longrightarrow B: k$

- A long 160-bit key  $k$  is used, but it can be transmitted over the Internet.
- Order of Messages 2 and 3 is not important.
- A  $b$ -bit authentication string optimises human work:

$$\text{Prob}[\text{a successful collision attack}] \approx 2^{-b}$$

- In message 1: node  $B$  is committed to the digest without knowing its value.

## Joint commitment without knowledge

$$1. \forall A \longrightarrow \forall A' : INFO_A, \text{Hash}(A, k_A)$$

- A group  $G$  of many parties want to exchange their authentic data  $INFO_A$ 's to one another over the insecure Dolev-Yao network.
- Each node  $A$  creates its own (160-bit) sub-key  $k_A$

## Joint commitment without knowledge

$$1. \forall A \longrightarrow \forall A' : INFO_A, \text{Hash}(A, k_A)$$

$$2. \forall A \longrightarrow \forall A' : k_A$$

- Each device  $A$  only reveals its (160-bit) sub-key  $k_A$  after  $A$  has received Messages 1 from all other nodes.
- All devices compute  $b$ -bit  $\text{digest}(k^*, INFO_S)$ , where
  - $k^* = k_A \oplus k_B \oplus k_C \oplus k_D \oplus \dots$
  - $INFO_S = INFO_A \parallel INFO_B \parallel INFO_C \parallel INFO_D \parallel \dots$

## Joint commitment without knowledge

1.  $\forall A \longrightarrow \forall A' : INFO_A, \text{Hash}(A, k_A)$

2.  $\forall A \longrightarrow \forall A' : k_A$

3.  $\forall A \Longrightarrow \forall A' : \text{all users compare } \text{digest}(k^*, INFOS)$

$$k^* = k_A \oplus k_B \oplus k_C \oplus \dots\dots\dots$$

$$INFOS = INFO_A \parallel INFO_B \parallel INFO_C \parallel \dots\dots$$

- Devices equally influence the final key  $k^*$ : they are *jointly committed* to the digest without knowing its value at the end of Messages 1.

## Joint commitment without knowledge

1.  $\forall A \longrightarrow \forall A' : INFO_A, \text{Hash}(A, k_A)$

2.  $\forall A \longrightarrow \forall A' : k_A$

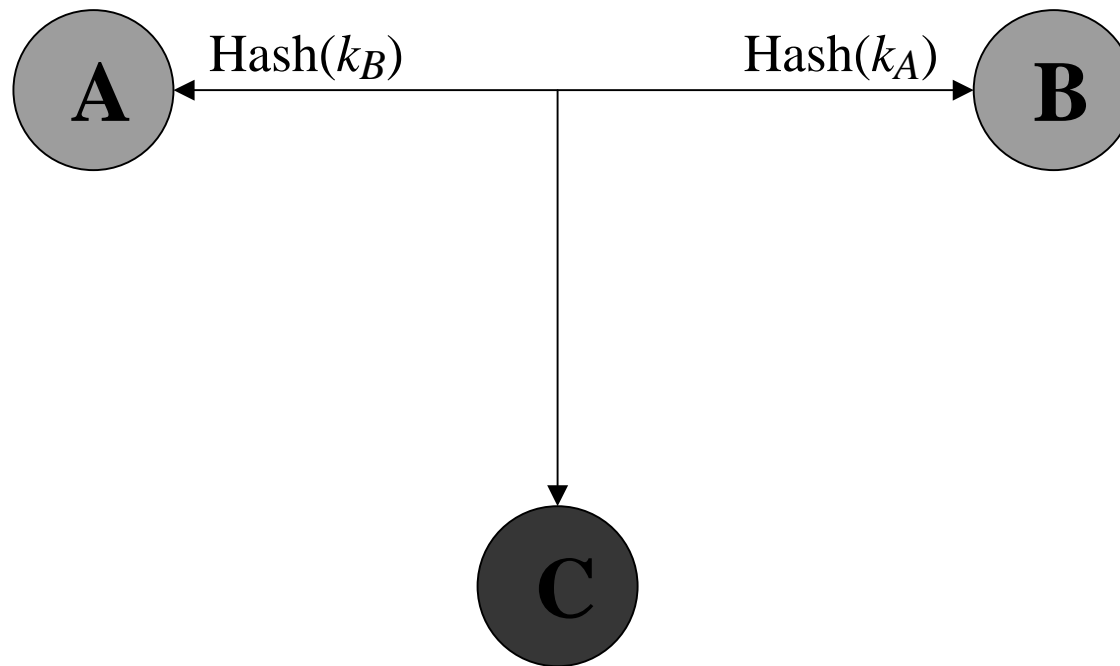
3.  $\forall A \Longrightarrow \forall A' : \text{all users compare } \text{digest}(k^*, INFOS)$

$$k^* = k_A \oplus k_B \oplus k_C \oplus \dots$$

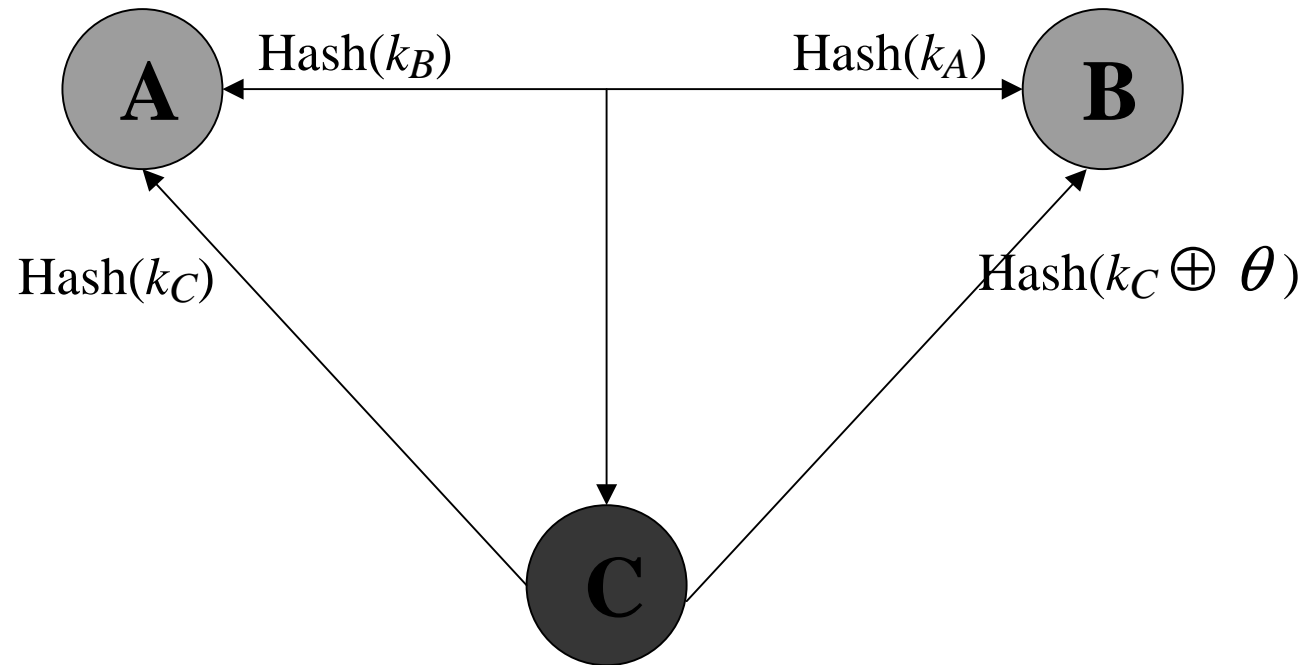
$$INFOS = INFO_A \parallel INFO_B \parallel INFO_C \parallel \dots$$

- Devices equally influence the final key  $k^*$ : they are *jointly committed* to the digest without knowing its value at the end of Messages 1.
- For any fixed  $\theta$ :  $\text{Prob}_k[\text{digest}(k, m_1) = \text{digest}(k \oplus \theta, m_2)] \leq \epsilon$

# Key manipulation in group protocol



## Key manipulation in group protocol



Party A:  $k^* = k_A \oplus k_B \oplus k_C$

Party B:  $k^* = k_A \oplus k_B \oplus k_C \oplus \theta$

# Security Model

- Parties compare a single string via *authentication channels*: physical or phone conversation, text messages, or manual data transfer between devices.
- All authentication strings are uniformly distributed and independent because they are randomised by random nonces of honest parties in each run.

# Security Model

- Parties compare a single string via *authentication channels*: physical or phone conversation, text messages, or manual data transfer between devices.
- All authentication strings are uniformly distributed and independent because they are randomised by random nonces of honest parties in each run.
- A successful ( $q$ -shot) attack means:
  - at least a pair of nodes compute the same authentication string, but
  - they receive different versions of data (INFO) that they try to agree on.

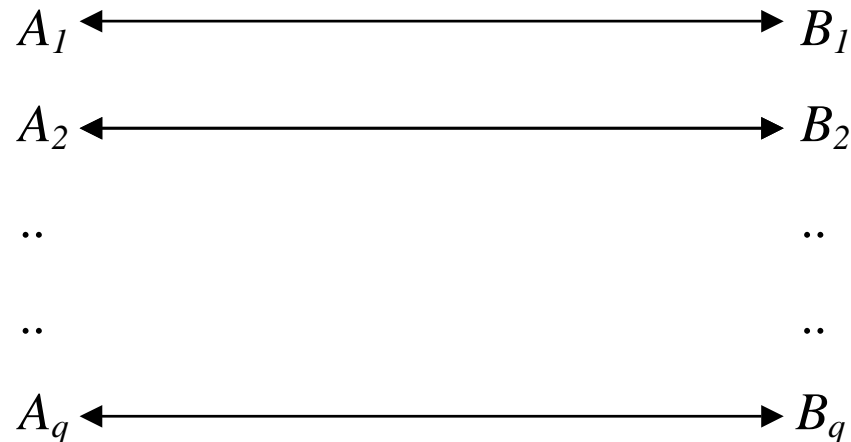
## *Strong* authentication channels

- All authentication strings are uniformly distributed and independent because they are randomised by random nonces of honest parties in each run.
- Authentication strings *cannot* be replayed and delayed:  
e.g. face-to-face conversation or manual data transfers.

## *Strong* authentication channels

- All authentication strings are uniformly distributed and independent because they are randomised by random nonces of honest parties in each run.
- Authentication strings *cannot* be replayed and delayed:  
e.g. face-to-face conversation or manual data transfers.

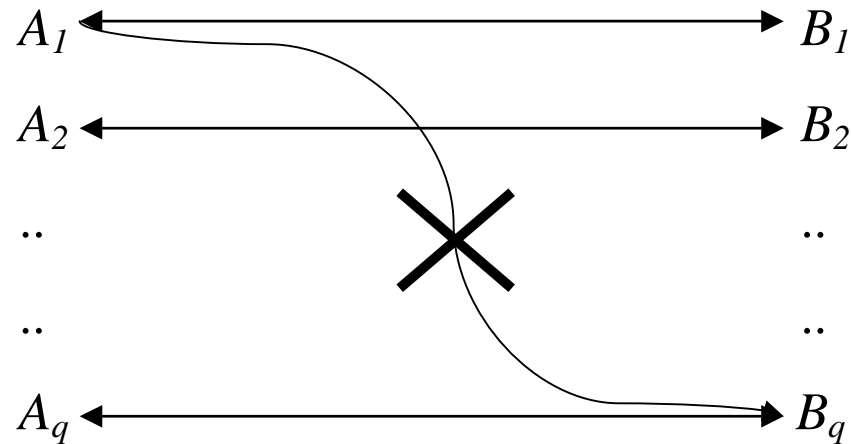
An intruder launches attacks on  $q$  runs of a protocol: a  $q$ -shot attack



## *Strong* authentication channels

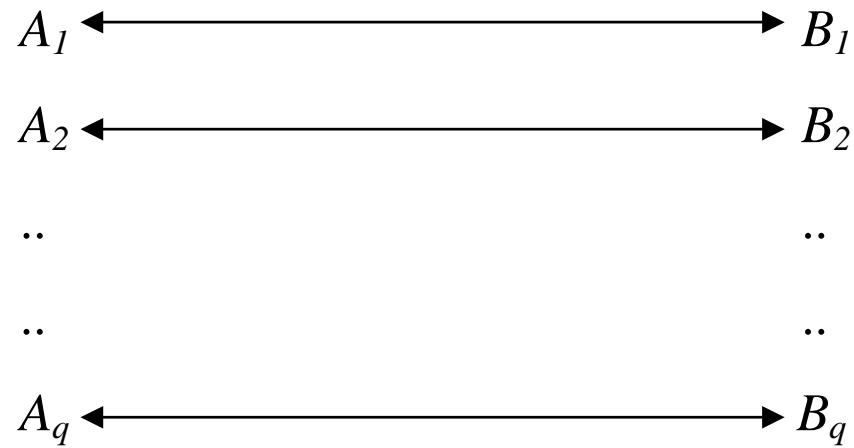
- All authentication strings are uniformly distributed and independent because they are randomised by random nonces of honest parties in each run.
- Authentication strings *cannot* be replayed and delayed:  
e.g. face-to-face conversation or manual data transfers.

An intruder launches attacks on  $q$  runs of a protocol: a  $q$ -shot attack



## *Strong* authentication channels

- All authentication strings are uniformly distributed and independent because they are randomised by random nonces of honest parties in each run.
- Authentication strings *cannot* be replayed and delayed:  
e.g. face-to-face conversation or manual data transfers.



$$\text{Prob}[q\text{-shot attack}] < q \times \text{Prob}[\text{one-shot attack}]$$

## Strength and Weakness of Security Model

- Strength: reduce the chance of a successful  $q$ -shot attack to a one-shot attack.

## Strength and Weakness of Security Model

- Strength: reduce the chance of a successful  $q$ -shot attack to a one-shot attack.
- Weakness: only accurate if attacks are on *different* pairs of parties, if not:
  - Humans might be aware or suspicious that attacks are taking place.
  - Humans allow no more attempt or require longer authentication strings.

## Strength and Weakness of Security Model

- Strength: reduce the chance of a successful  $q$ -shot attack to a one-shot attack
- Weakness: only accurate if attacks are on *different* pairs of parties, if not:
  - Humans might be aware or suspicious that attacks are taking place.
  - Humans allow no more attempt or require longer authentication strings.
- Extending the authentication string by 1 bit after each mismatch makes

$$\text{Prob}[q\text{-shot attack}] = 2^{-b} + 2^{-(b+1)} + \dots + 2^{-(b+q-1)} < 2^{1-b}$$

## *Weak* authentication channels

[Vaudenay, CRYPTO 2005]

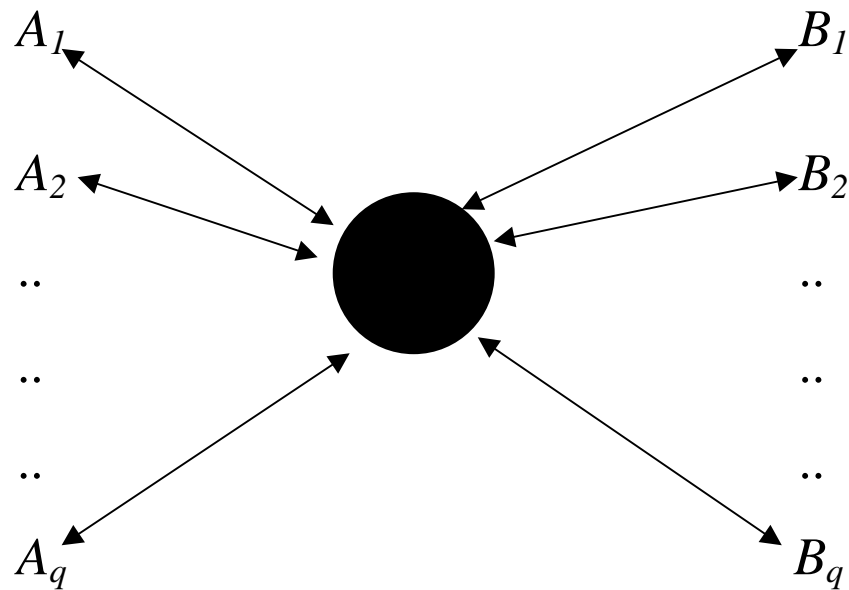
- All authentication strings are uniformly distributed and independent because they are randomised by random nonces of honest parties in each run.
- Authentication strings *can* be replayed and delayed:  
e.g. telephone conversation or text messages.

# *Weak* authentication channels

[Vaudenay, CRYPTO 2005]

- All authentication strings are uniformly distributed and independent because they are randomised by random nonces of honest parties in each run.
- Authentication strings *can* be replayed and delayed:  
e.g. telephone conversation or text messages.

An intruder launches attacks on  $q$  protocol runs between the *same* pair of nodes  $(A,B)$ :  
a  $q$ -shot attack

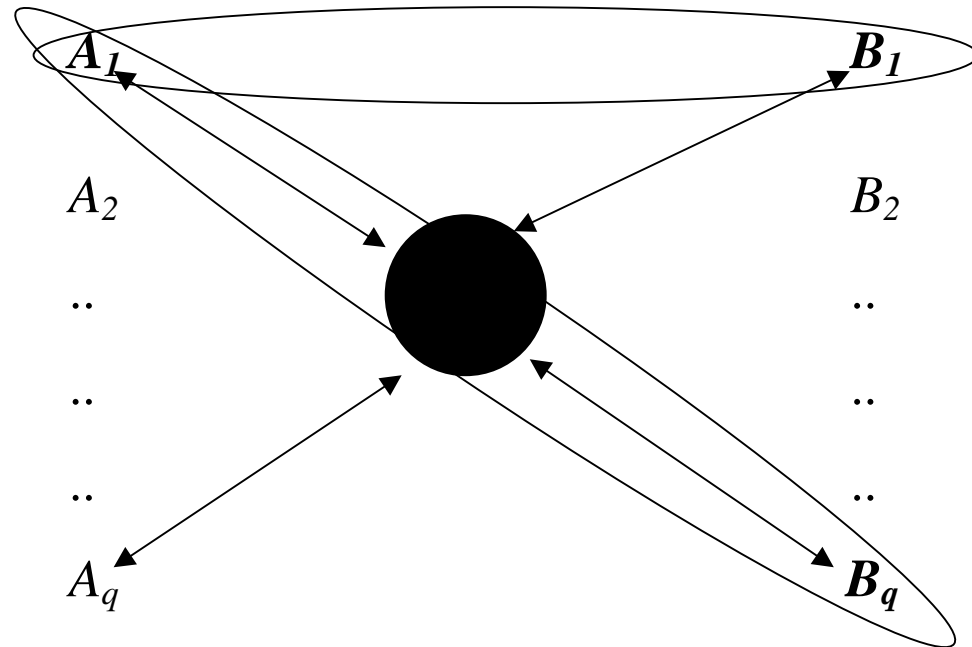


# *Weak* authentication channels

[Vaudenay, CRYPTO 2005]

- All authentication strings are uniformly distributed and independent because they are randomised by random nonces of honest parties in each run.
- Authentication strings *can* be replayed and delayed:  
e.g. telephone conversation or text messages.

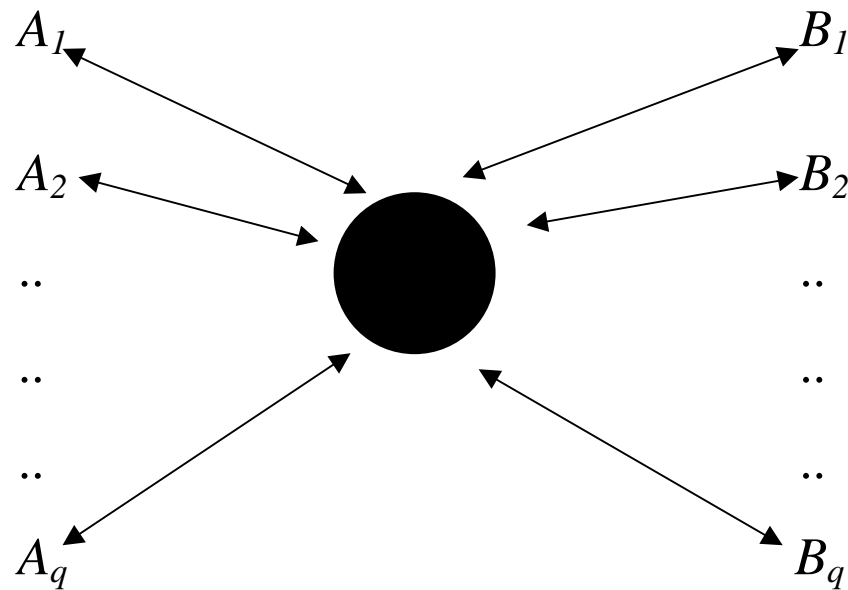
An intruder launches attacks on  $q$  protocol runs between the *same* pair of nodes  $(A,B)$ :  
a  $q$ -shot attack



# *Weak* authentication channels

[Vaudenay, CRYPTO 2005]

- All authentication strings are uniformly distributed and independent because they are randomised by random nonces of honest parties in each run.
- Authentication strings *can* be replayed and delayed:  
e.g. telephone conversation or text messages.



$$\text{Prob}[q\text{-shot attack}] < q^2 \times \text{Prob}[\text{one-shot attack}]$$

# Summary

- New authentication technology based on human trust and human interaction can reduce the need for passwords and PKI.
- Exploit properties of human interactions to restrict the power of the adversary to prove the security of protocols.

Many thanks for your attention.

For more information:

- My website: <http://web.comlab.ox.ac.uk/people/Long.Nguyen/>
- L.H. Nguyen and A.W. Roscoe. *Authentication protocols based on low-bandwidth unspoofable channels: a comparative survey*. Submitted to Journal of Computer Security. Manuscript is available on website.
- L.H. Nguyen and A.W. Roscoe. *Authenticating ad hoc networks by comparison of short digests*. Information and Computation 206 (2008), 250-271. And Proceedings of FCS-ARSPA 2006.
- L.H. Nguyen and A.W. Roscoe. *Separating two roles of hashing in one-way message authentication*. Proceedings of FCS-ARSPA-WITS 2008.
- L.H. Nguyen and A.W. Roscoe. *New combinatorial bounds for universal families of hash functions*. Submitted to SAC'09, copy is available on website.

## Toeplitz matrix multiplication: digest functions

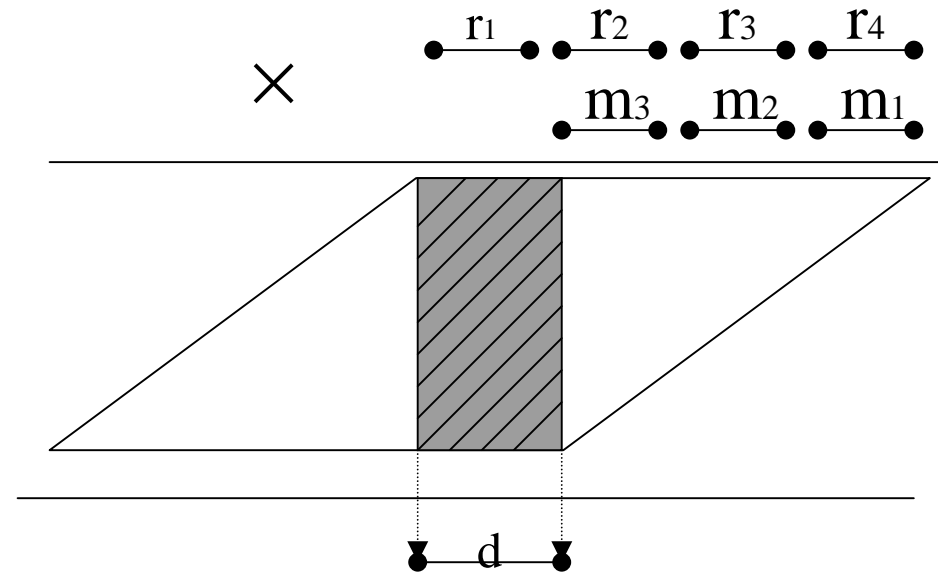
- Definition of a Toeplitz matrix  $R$ :  $\forall (i,j), R_{i,j} = R_{i+1,j+1}$
- We derive  $b + |m| - 1$  pseudorandom bits from key  $k$  (e.g. via LFSR) to construct a Toeplitz matrix  $R$  of  $b$  columns and  $|m|$  rows.

$$\text{digest}(k,m) = m \bullet R \pmod{2}$$

- This construction is due to Krawczyk (CRYPTO'94, LNCS vol.839).
- Toeplitz matrix multiplication can be replaced by integer multiplication.

# Integer multiplication: digest function

- We are only interested in the overlap of integer multiplication.
- The overlap has similarities to Toeplitz matrix.
- But multiplication of long messages is expensive.

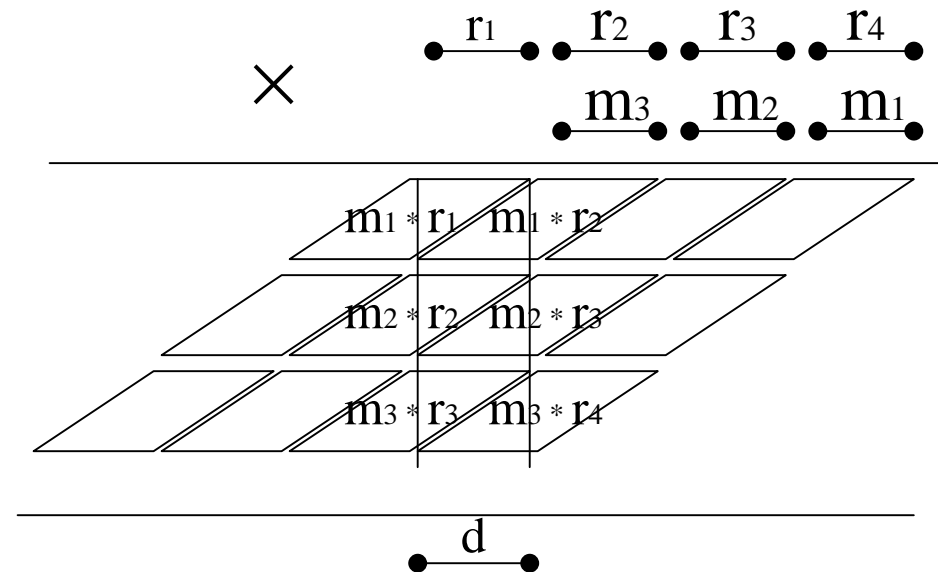


$$m = m_3 \parallel m_2 \parallel m_1$$

$$\text{PRNG}(k) = r_1 \parallel r_2 \parallel r_3 \parallel r_4$$

# Integer multiplication: digest function

- $b$ -bit (16-32) integer multiplications are fast.
- We are only interested in the overlap area.
- The impact of carry-bits is reduced.



$$\text{PRNG}(k) = r_1 \parallel r_2 \parallel r_3 \parallel r_4$$

$$m = m_3 \parallel m_2 \parallel m_1$$

$$\begin{aligned} \text{digest}(k,m) = & \text{Low}(m_1 * r_1) + \text{Up}(m_1 * r_2) + \\ & \text{Low}(m_2 * r_2) + \text{Up}(m_2 * r_3) + \\ & \text{Low}(m_3 * r_3) + \text{Up}(m_3 * r_4) \end{aligned} \quad 39$$

## *digest* function: security and efficiency

- $b$ -bit output *digest* function's specification

As key  $k$  varies uniformly:

- for any fixed message  $m$ ,  $\text{digest}(k, m)$  is uniformly distributed.
- for any fixed  $m_1 \neq m_2$ :  $\text{Prob}_k[\text{digest}(k, m_1) = \text{digest}(k, m_2)] \leq \epsilon$

- *digest* has a short output (e.g.  $b = 16$  bits): will probably be (significantly) faster to compute than cryptographic hashes: 160-bit SHA, MD5.

## *digest* function: security and efficiency

- $b$ -bit output *digest* function's specification

As key  $k$  varies uniformly:

- for any fixed message  $m$ ,  $\text{digest}(k, m)$  is uniformly distributed.
- for any fixed  $m_1 \neq m_2$ :  $\text{Prob}_k[\text{digest}(k, m_1) = \text{digest}(k, m_2)] \leq \epsilon$

- *digest* has a short output (e.g.  $b = 16$  bits): will probably be (significantly) faster to compute than cryptographic hashes: 160-bit SHA, MD5.
- Asymptotic computation complexity of hash functions in practice, i.e. Merkle-Damgård construction and the avalanche effect.

$\text{Cost}(\text{digest/hash}) \approx \text{message-input-length} \times \text{hash-output-length}$

## New applications of *efficient* digests

- Conventional signature:  $M \parallel \text{Sign}_A(\text{Hash}(M))$   
For large  $M$  (DVD), hashing time overtakes signing and other operators.
- Replace cryptographic hash with short digest to sign faster.

## Digital signatures with efficient digest

- Conventional signature:  $M \parallel \text{Sign}_A(\text{Hash}(M))$   
For large  $M$  (DVD), hashing time overtakes signing and other operators.
- Replace cryptographic hash with short digest to sign faster.

$$A \longrightarrow B \left\{ \begin{array}{l} M, \text{digest}(k, M) \\ \text{Sign}_A(\{B, k\}_{pk(B)}) \end{array} \right.$$

