SECRYPT 2016
13th International Conference on Security and Cryptography
26 - 28 July, 2016
Lisbon, Portugal
ICETE 2016

User-friendly Manual Transfer of Authenticated Online Banking Transaction Data

Sven Kiljan, Harald Vranken & Marko van Eekelen

sven@kiljan.org

www.kiljan.org

July 26, 2016

NHL HOGESCHOOL

Open Universiteit

Radboud University

Introduction

- Sven Kiljan

- PhD student at Open University of the Netherlands
- Daily work at iCIS Digital Security group at Radboud University

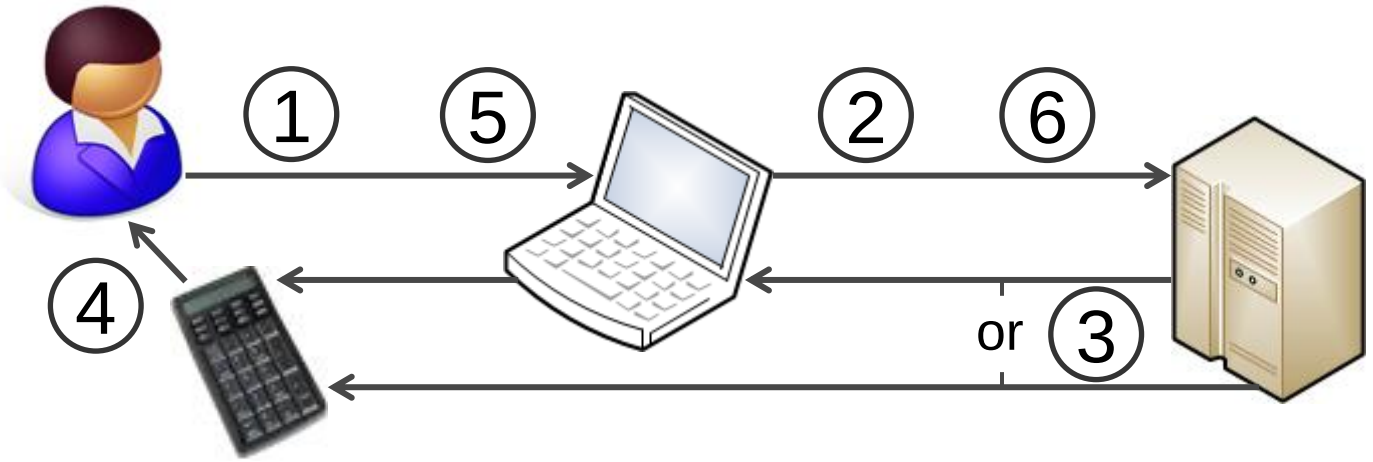
- Research: improving technical security in online banking

Presentation contents

- Description of online banking transaction authorization schemes
  - Used by banks: About What You **See** Is What You Sign
  - Our (previous) proposal: About What You **Enter** Is What You Sign
- Our current proposal: a Message Code to transfer critical transaction information
- Discussion and limitations
- Questions

What You See Is What You Sign

- An authorization scheme that allows users to securely verify information received earlier by banks



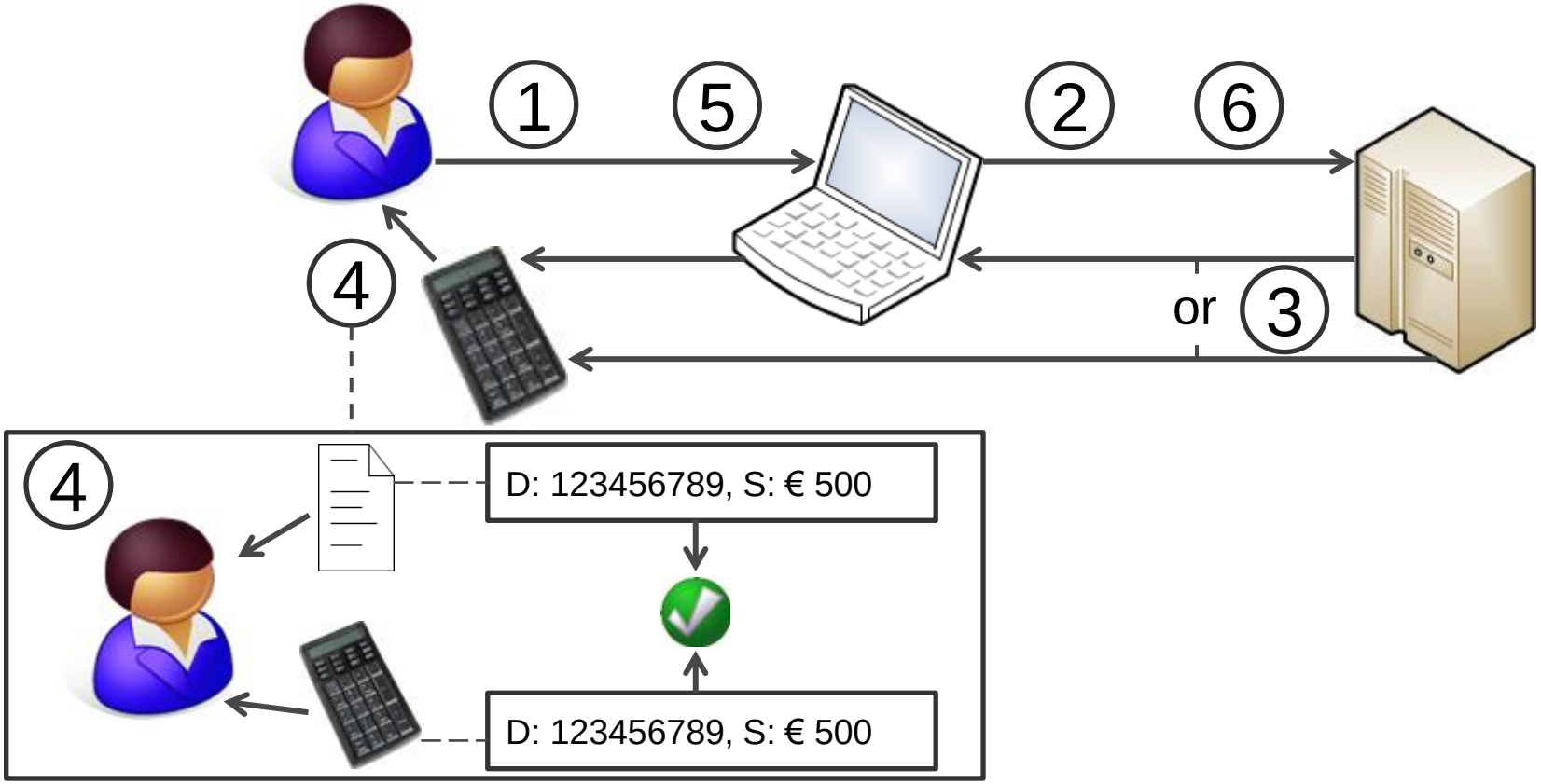
1 2 Critical transaction information (insecure)

3 4 Critical transaction information and confirmation code (secure)

5 6 Confirmation code (one-time password, so secure)

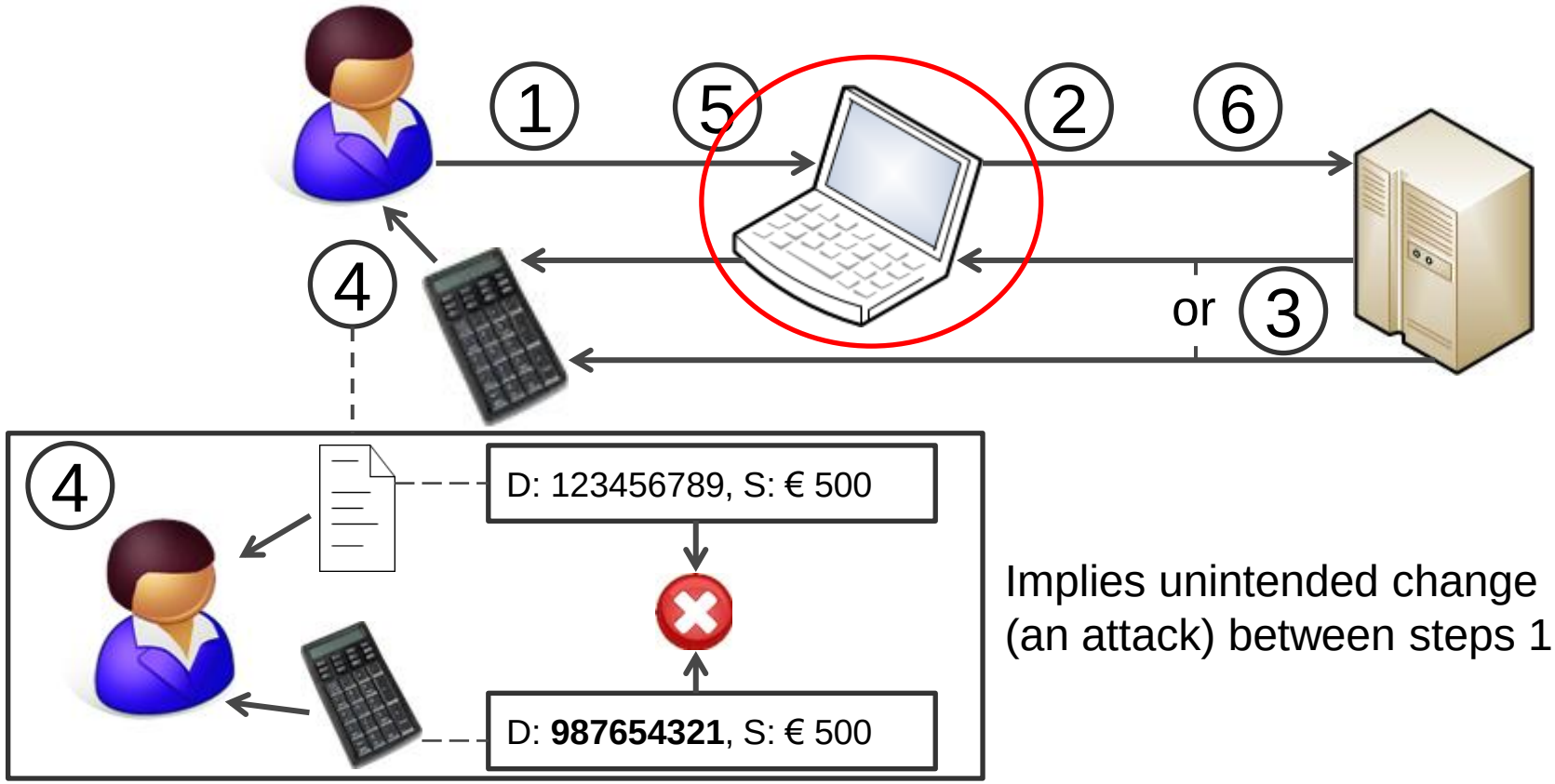
What You See Is What You Sign

- An authorization scheme that allows users to securely verify information received earlier by banks



What You See Is What You Sign

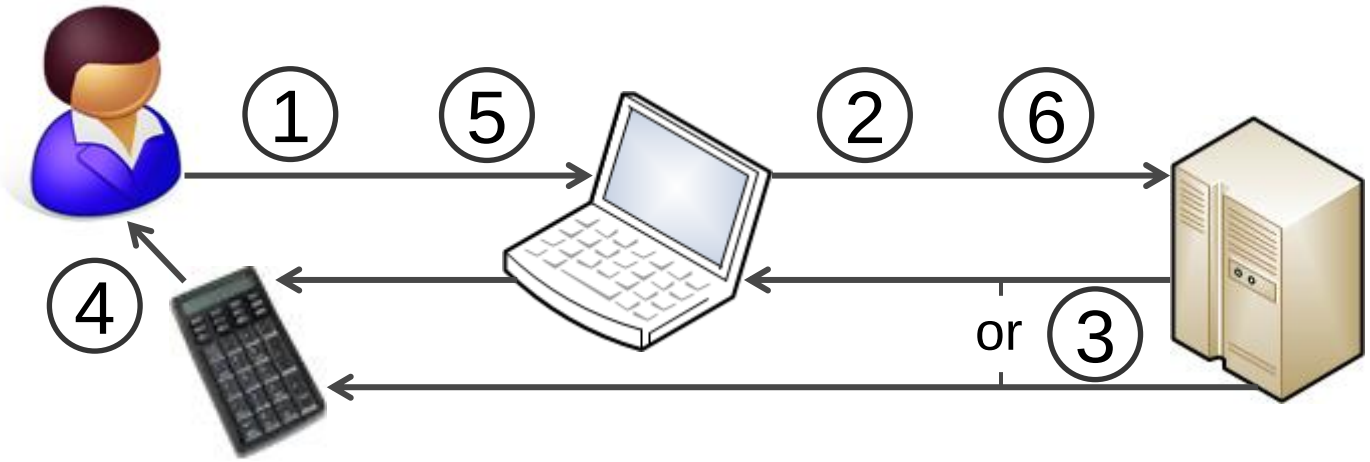
- An authorization scheme that allows users to securely verify information received earlier by banks



Implies unintended change
(an attack) between steps 1 and 2.

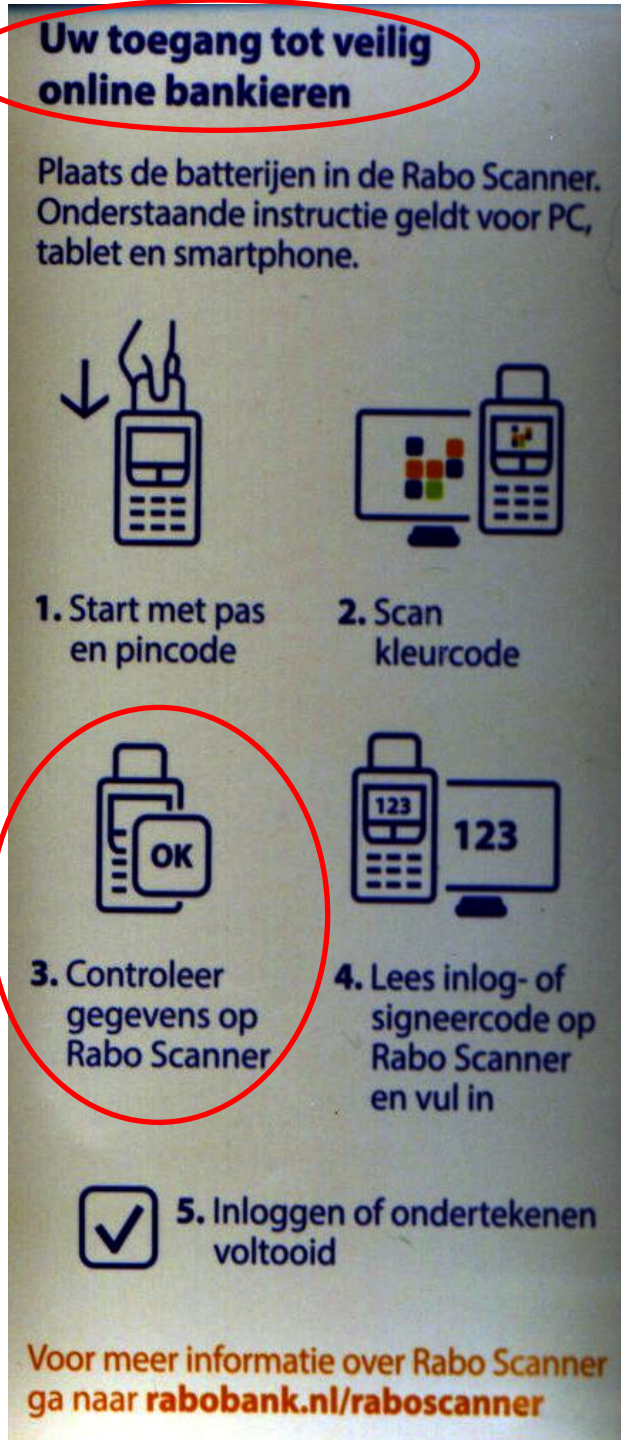
What You See Is What You Sign
How secure is it?

- An authorization scheme that allows users to securely verify information received earlier by banks



- 21% of attacks against the user's ability to recognize significantly changed account numbers succeed (AlZomai et al., 2008)
- From a usability perspective, this is not usable. Therefore, it is not secure at all!

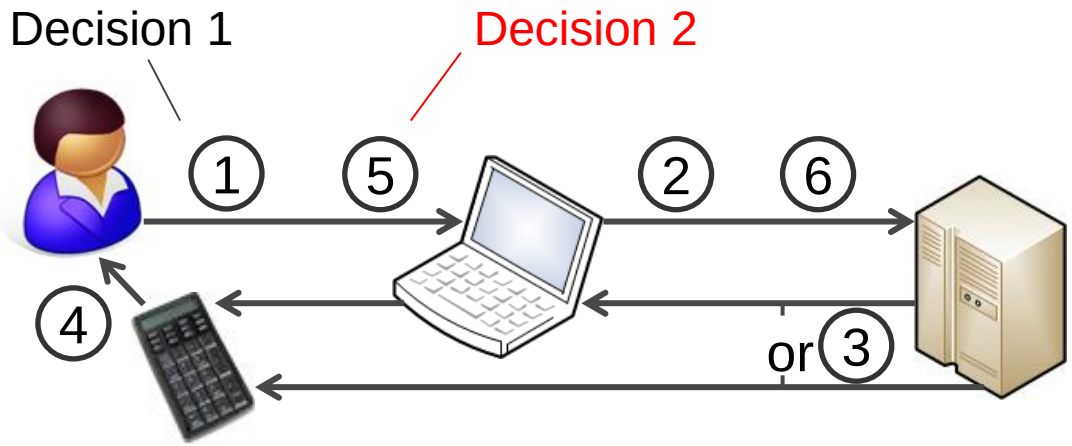
What You See Is What You Sign



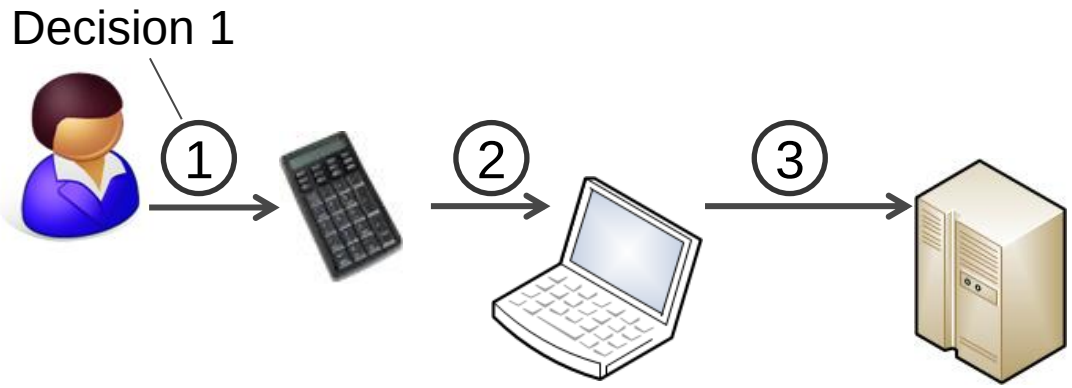
User-friendly Manual Transfer of Authenticated Online Banking Transaction Data

Introducing What You Enter Is What You Sign

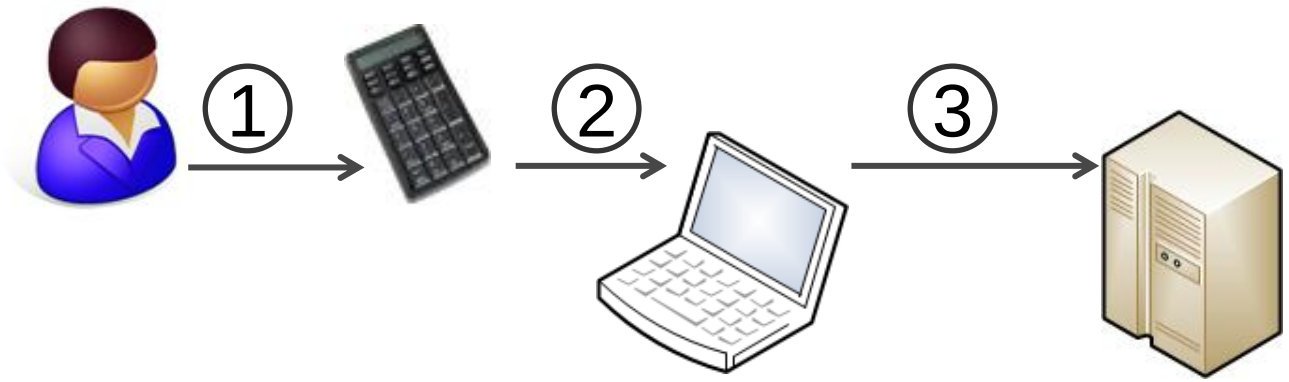
- The natural way to use a system should be the secure way (Yee, 2002), so this is not secure:



- A user expresses his or her desire to create a transaction by entering information in a system (the natural way):



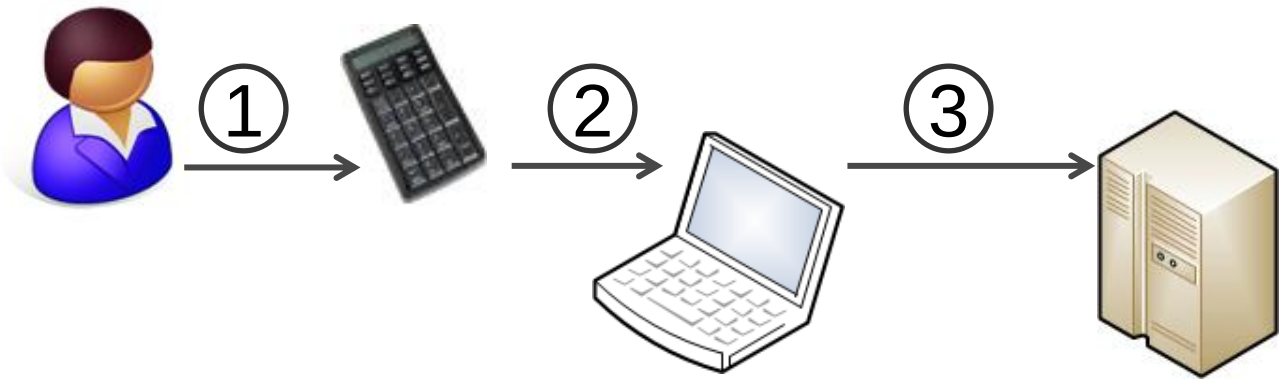
The initial proposal for What You Enter Is What You Sign



1 Critical transaction information (entry)

2 3 Critical transaction information (secured by signature)

The initial proposal for What You Enter Is What You Sign



- Some limitations. The biggest one: connectivity
- We proposed keyboard emulation
  - USB: not all devices have USB connectivity, and those which do can have different physical connectors (USB A, micro-USB (OTG), USB Type C...)
  - Bluetooth: not all devices have Bluetooth, needs initial pairing (not user-friendly) and an active radio (battery drain) when used

Presentation contents

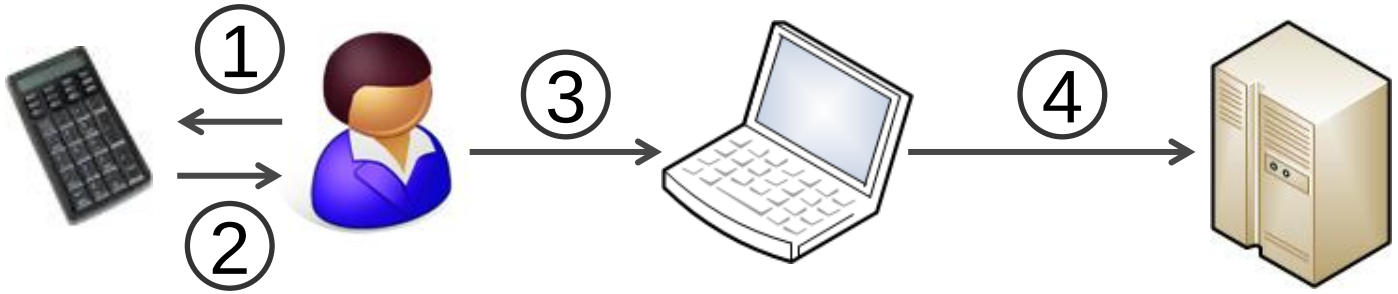
- Description of online banking transaction authorization schemes
  - Used by banks: About What You **See** Is What You Sign
  - Our (previous) proposal: About What You **Enter** Is What You Sign

- Our current proposal: a Message Code to transfer critical transaction information

- Discussion and limitations

- Questions

Our new proposal from our paper



1 Critical transaction information

2 3 4 Message Code (secured by embedded signature)

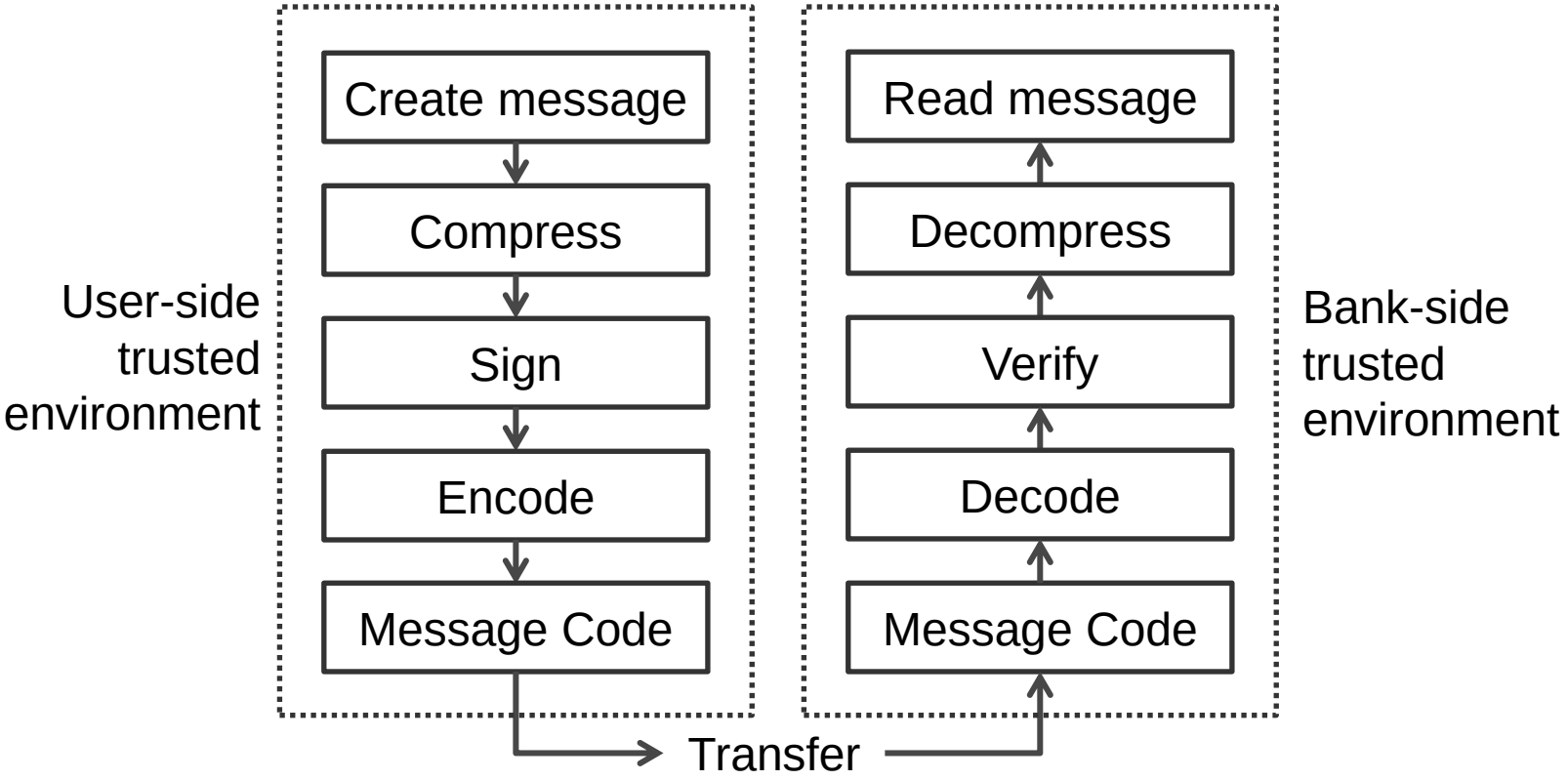
- Message code contains critical transaction information and a cryptographic signature
- Similar to entering a one-time password or a response in challenge-response authentication: does not require deep cognitive skills
- No electronic connection: reduced attack surface

Creating a Message Code – Requirements

- Security requirements
  - Provide authenticity
  - Provide integrity
  - Confidentiality is not important

- Usability requirements
  - Easy to read (no O and 0)
  - Easy entry (no ` ~ ! @ # $ % ^ & * ( ) - = _ + [ ] { } \ | ; ’ : ” , . < > / ?)
  - As short as possible

Creating a Message Code – Process

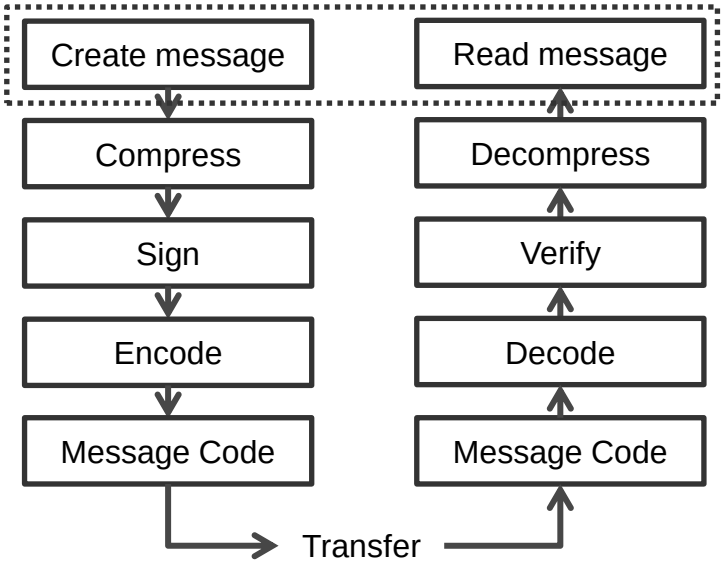


Creating a Message Code – Message

- Message Code should be as short as possible, so only critical information

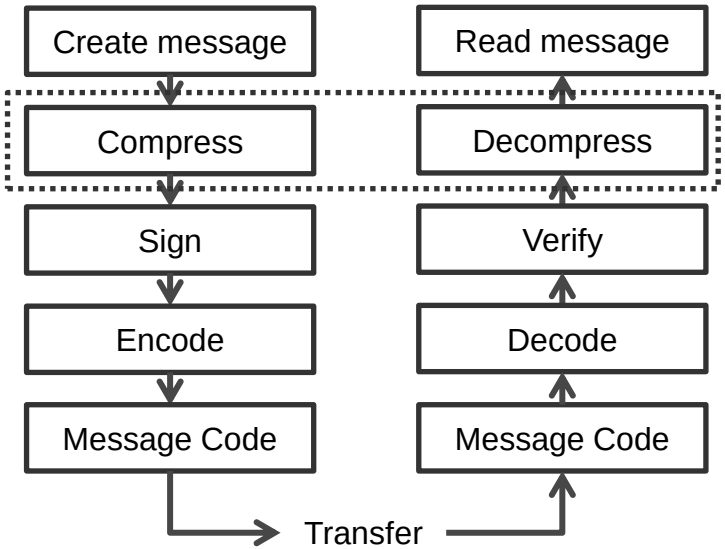
- Destination bank account number
  - In the Netherlands, so an IBAN string
  - Example: NL76 SIMB 0759 5958 79

- Amount
  - Currency and decimal value
  - Example: € 123456.78



Creating a Message Code – Compression

- Shorter message = Shorter Message Code
- Assumption: only domestic transfers

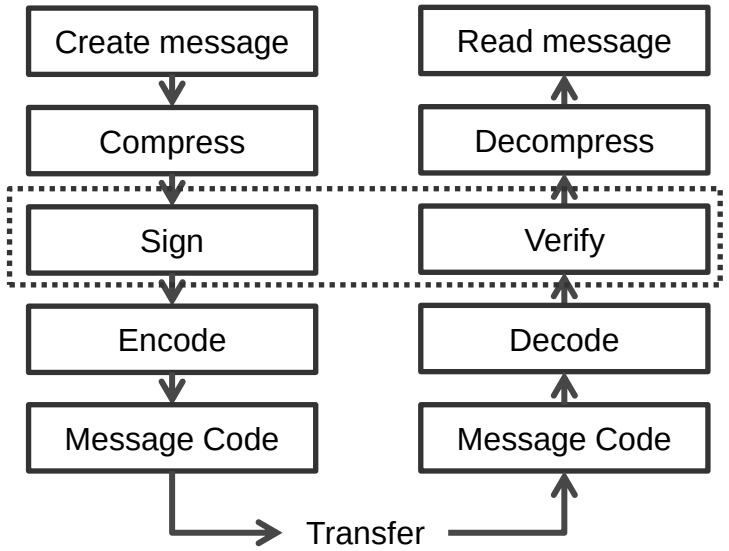


- Dutch IBAN
  - Original (string): NL76 SIMB 0759 5958 79
  - Compressed (integer): 759595879

- Amount
  - Original (currency and decimal number) : € 123456.78
  - Compressed (integer): 12345678

- Raw data: “759595879 12345678”

Creating a Message Code – Signing

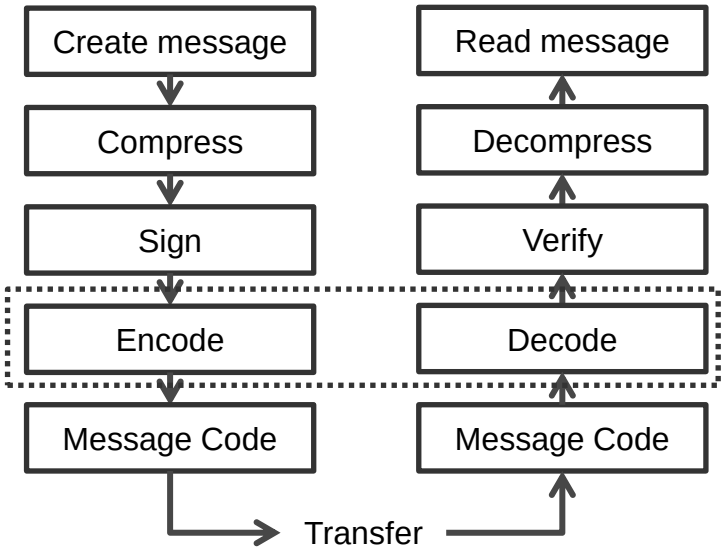


- Secures authenticity and integrity
- We picked a very simple implementation, using a MAC: *mac* = H(*k* | *n* | *m*)
- H – SHA-3 (Keccak) hash function
- *k* – randomly generated secret key
- *n* – nonce (must be predictable by bank, since it will not be part of the Message Code)
- *m* – message

- Example data, where *k* = 0x00, 0x01... 0xFF and *n* = 0x00 0x00 0x00 0x01 and *m* = 759595879 (30 bits) concatenated with 12345678 (27 bits)
- *mac* = 0x77 0x64 0xCE (23 bits)

Creating a Message Code – Encoding

- Convert raw data to human readable and writable text, using Z-Base 32



z-base-32 alphabet

Value	Symbol	Value	Symbol	Value	Symbol	Value	Symbol
0	y	8	e	16	o	24	a
1	b	9	j	17	t	25	3
2	n	10	k	18	1	26	4
3	d	11	m	19	u	27	5
4	r	12	c	20	w	28	h
5	f	13	p	21	i	29	7
6	g	14	q	22	s	30	6
7	8	15	x	23	z	31	9

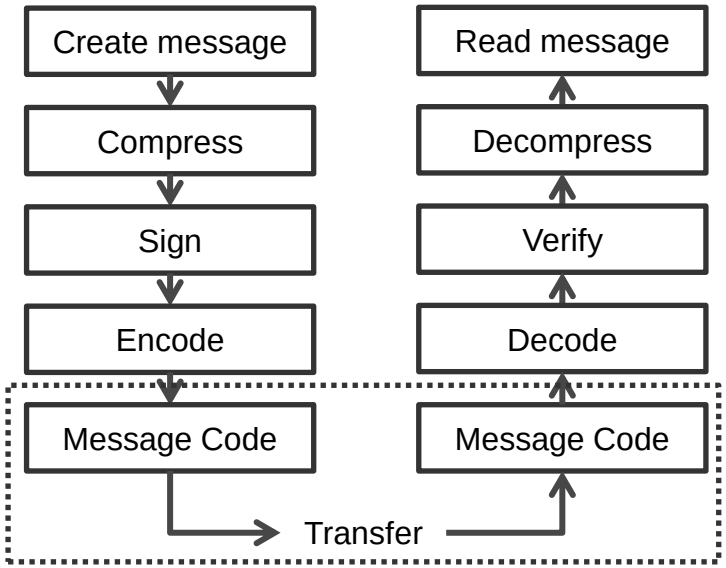
	Compressed message														
	Account number (30 bits): 759595879						Amount (27 bits): 12345678						MAC (23 bits):0x7764ce		
Binary	10110	10100	01101	00000	11011	00111	00010	11110	00110	00010	10011	10111	01110	11001	00110	01110
Decimal	22	20	13	0	27	7	2	30	6	2	19	23	14	25	6	14
Z-Base32	s	w	p	y	5	8	n	6	g	n	u	z	q	3	g	q

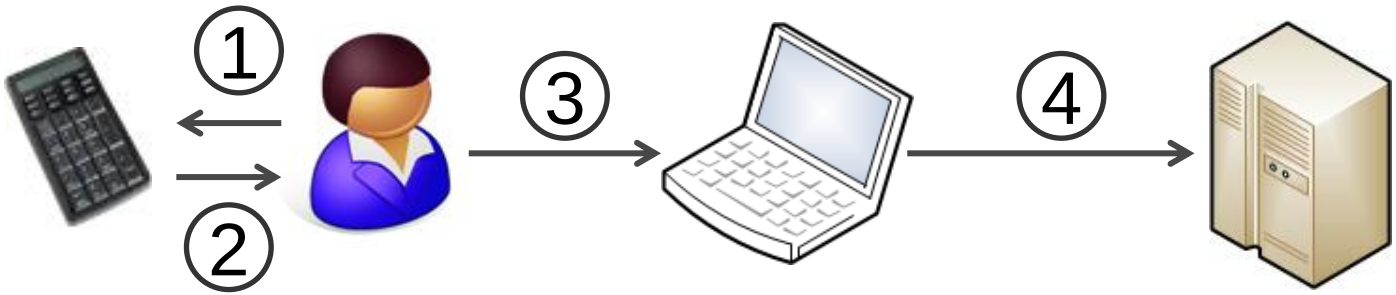
Transfer the Message Code

- Resulting code:

  `swpy-58n6-gnuz-q3gq`

- Possible implementation
  - User enters text in their computer





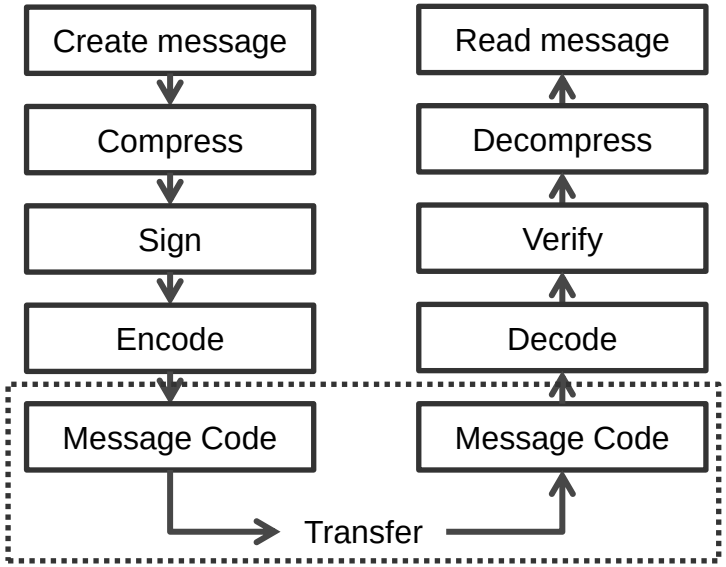
Transfer the Message Code

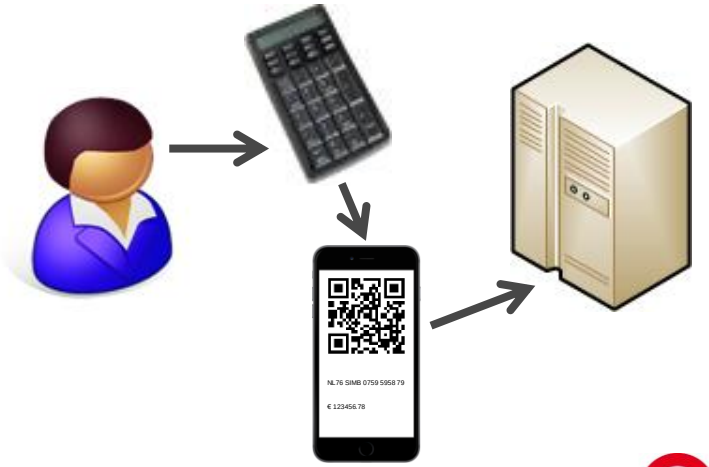
- Resulting code:

  `swpy-58n6-gnuz-q3gq`

- Possible implementation
  - Alternative for mobile banking:

    the code is shown as a QR code





Presentation contents

- Description of online banking transaction authorization schemes
  - Used by banks: About What You **See** Is What You Sign
  - Our (previous) proposal: About What You **Enter** Is What You Sign
- Our current proposal: a Message Code to transfer critical transaction information
- Discussion and limitations
- Questions

Discussion and limitations

- What You Enter Is What You Sign is only applicable when users provide transaction information.

- The signature length is short (only 23 bits in this example), but brute force attempts can be limited since only the bank is capable of determining whether a signature is valid. Too many attempts in a short period of time can result in an account lockdown.

- A limitation (that also applies to What You See Is What You Sign) is the confirmation a bank sends to the user after a transaction is successfully processed. An attacker can change this to a message that indicates a failure in the hope that the user will redo the same transaction again.

Questions?

Thank you for your attention!

- AlZomai, M., AlFayyadh, B., Jøsang, A., and McCullagh, A. (2008). An exprimental investigation of the usability of transaction authorization in online bank security systems. In *Proceedings of the sixth Australasian conference on Information security-Volume 81*, pages 65–73. Australian Computer Society, Inc.

- Yee, K.-P., 2002. User Interaction Design for Secure Systems. In Deng, R., Bao, F., Zhou, J., Qing, S. (Eds.), *Information and Communications Security*. Vol. 2513 of Lecture Notes in Computer Science, pages 278-290. Springer Berlin Heidelberg.