

Next Generation Crypto

Light-weight crypto.

Quantum-robust crypto

Tokenization.

Zero-knowledge.

Homomorphic Encryption.

zkSnarks, Range-proofs

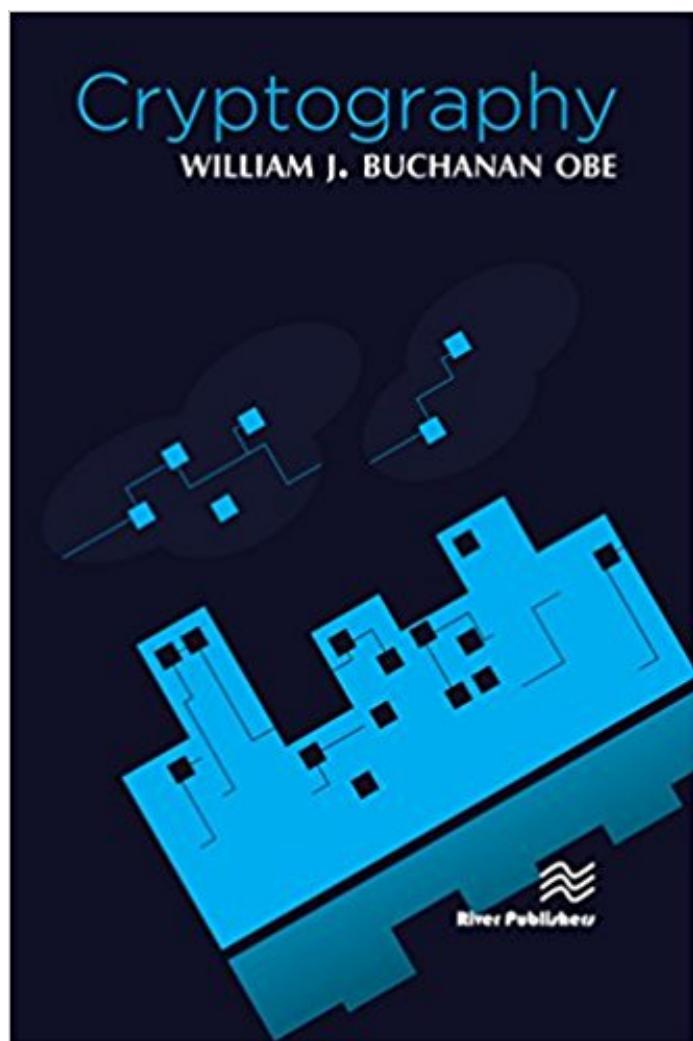


Prof Bill Buchanan OBE

<http://asecuritysite.com/encryption>

Future Crypto

Light-weight crypto

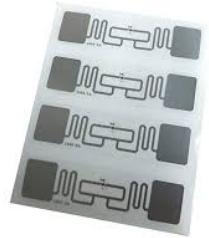


A Computer?



Light-weight crypto

- Conventional cryptography. Servers and Desktops. Tablets and smart phones.
- Light-weight cryptography. Embedded Systems. RFID and Sensor Networks.



Thus often light-weight cryptography methods balance performance (throughput) against **power drain and GE (gate equivalents)**. Along with this the method must also have a low requirement for RAM (where the method requires the usage of running memory to perform its operation) and ROM (where the method is stored on the device). In order to assess the strengths of various methods we often **define the area** that the cryptography function will use on the device — and which is defined in μm^2 .

[View methods](#)

Light-weight crypto

Cipher	Key bits	Block bits	Cycles per block	Throughput at 100 kHz (Kbps)	Logic process	Area (GEs)
Block ciphers						
Present	80	64	32	200.00	0.18 μm	1,570
AES	128	128	1,032	12.40	0.35 μm	3,400
Hight	128	64	34	188.20	0.25 μm	3,048
Clefia	128	128	36	355.56	0.09 μm	4,993
mCrypton	96	64	13	492.30	0.13 μm	2,681
DES	56	64	144	44.40	0.18 μm	2,309
DESXL	184	64	144	44.40	0.18 μm	2,168
Stream ciphers						
Trivium ⁵	80	1	1	100.00	0.13 μm	2,599
Grain ⁵	80	1	1	100.00	0.13 μm	1,294

*AES: Advanced Encryption Standard; DES: Data Encryption Standard; DESXL: lightweight DES with key whitening.

function will use on the device — and which is defined in μm^2 .

[View methods](#)

Light-weight crypto

Cipher	Key size (bits)	Block size (bits)	Encryption (cycles/block)	Throughput at 4 MHz (Kbps)	Decryption (cycles/block)	Relative throughput (% of AES)	Code size (bytes)	SRAM size (bytes)	Relative code size (% of AES)
Hardware-oriented block ciphers									
DES	56	64	8,633	29.6	8,154	38.4	4,314	0	152.4
DESSL	184	64	8,531	30.4	7,961	39.4	3,192	0	112.8
Hight	128	64	2,964	80.3	2,964	104.2	5,672	0	200.4
Present	80	64	10,723	23.7	11,239	30.7	936	0	33.1
Software-oriented block ciphers									
AES	128	128	6,637	77.1	7,429	100.0	2,606	224	100.0
IDEA	128	64	2,700	94.8	15,393	123.0	596	0	21.1
TEA	128	64	6,271	40.8	6,299	53.0	1,140	0	40.3
SEA	96	96	9,654	39.7	9,654	51.5	2,132	0	75.3
Software-oriented stream ciphers									
Salsa20	128	512	18,400	111.3	NA	144.4	1,452	280	61.2
LEX	128	320	5,963	214.6	NA	287.3	1,598	304	67.2

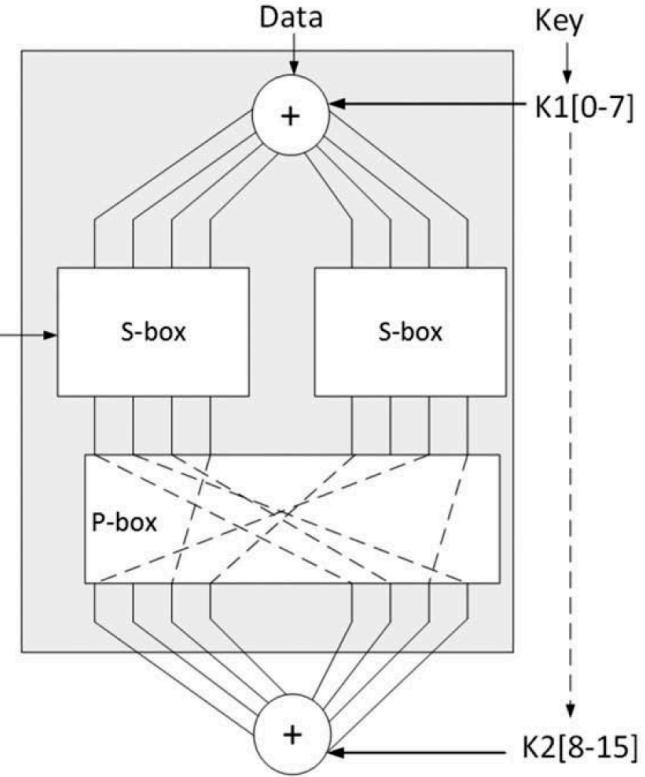
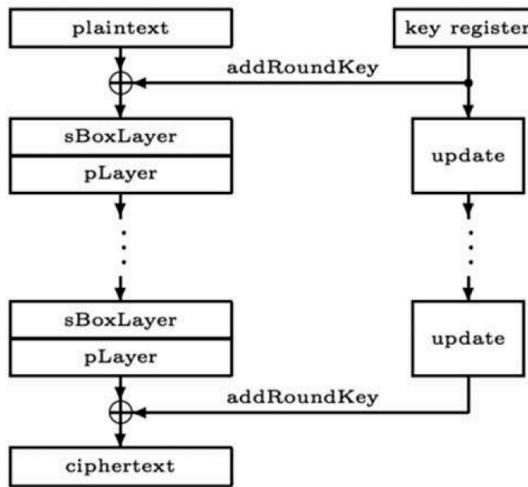
*IDEA: International Data Encryption Algorithm; TEA: Tiny Encryption Algorithm; SEA: Scalable Encryption Algorithm.

function will use on the device and which is defined in [pinout](#).

[View methods](#)

Light-weight crypto (PRESENT)

```
generateRoundKeys()  
for i = 1 to 31 do  
    addRoundKey(STATE, $K_i$ )  
    sBoxLayer(STATE)  
    pLayer(STATE)  
end for  
addRoundKey(STATE, $K_{32}$ )
```

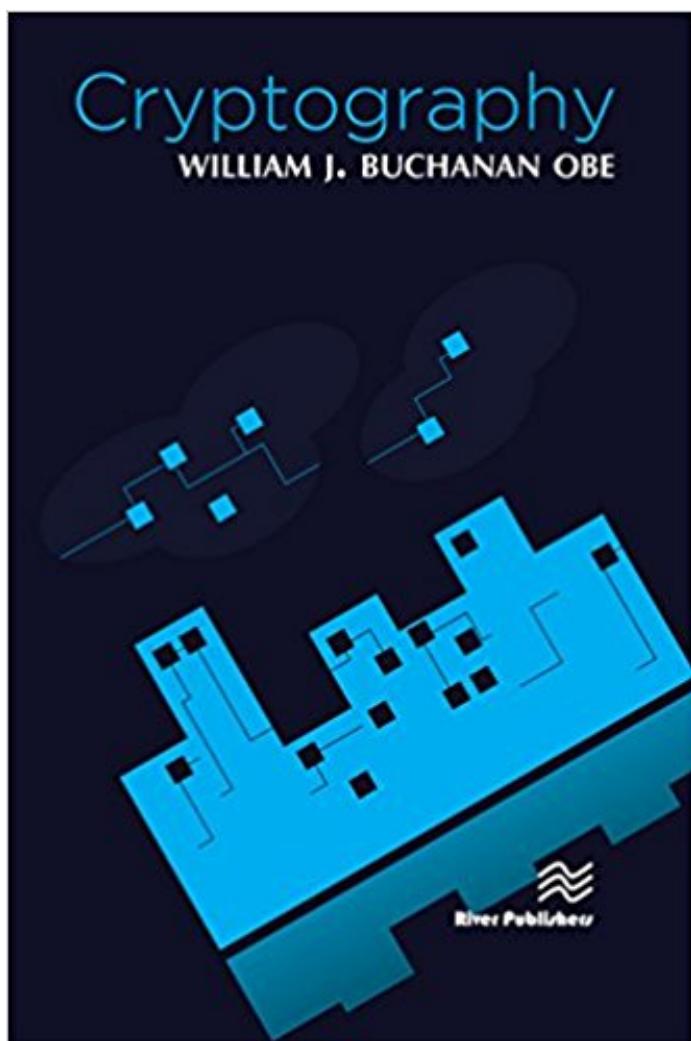


x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S[x]$	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

[View methods](#)

Future Crypto

Quantum-robust crypto

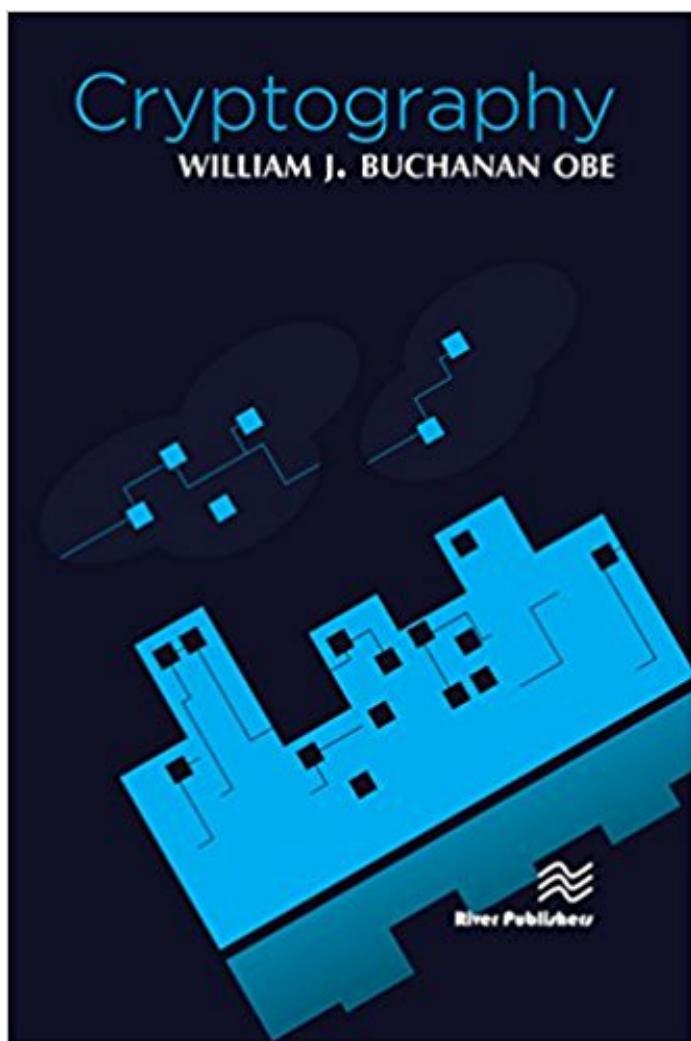


Quantum-robust crypto

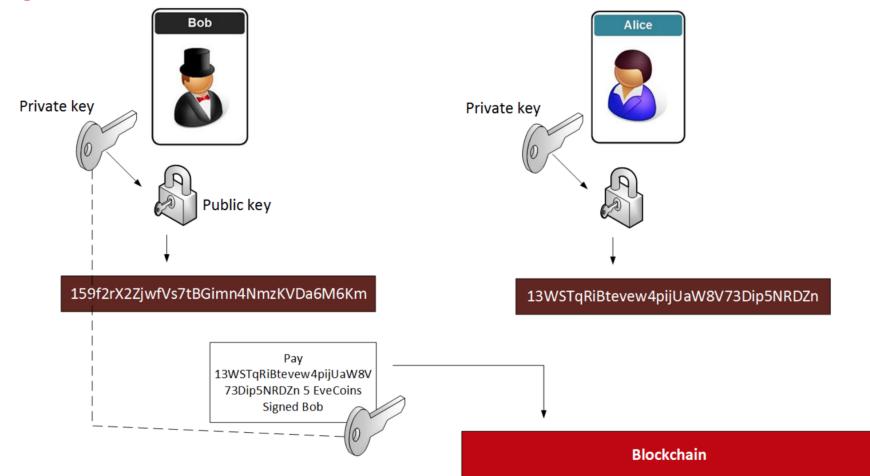
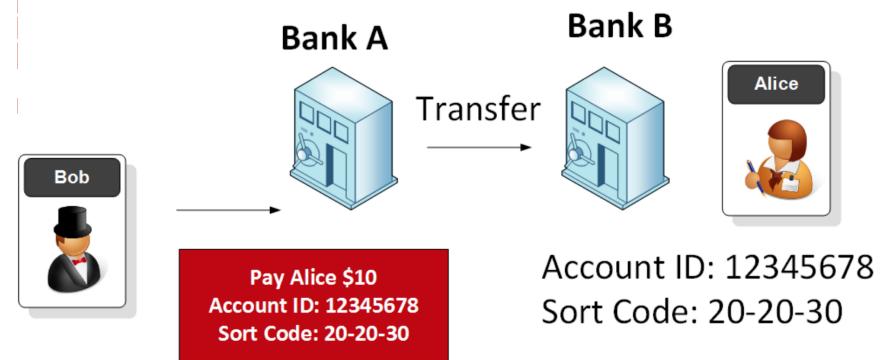
- **Lattice-based cryptography** [[Lattice](#)] — This classification shows great potential and is leading to new cryptography methods, such as for fully homomorphic encryption, and code obfuscation.
- **Code-based cryptography** [[McEliece](#)] — This classification was created in 1978 with the McEliece cryptosystem but has barely been using in real applications. The McEliece method uses linear codes that are used in error correcting codes, and involves matrix-vector multiplication. An example of a linear code is Hamming code.
- **Multivariate polynomial cryptography** [[UOV](#)] — This classification involves the difficulty of solving systems of multivariate polynomials over finite fields. Unfortunately, many of the methods that have been proposed have already been broken.
- **Hash-based signatures** [[GMSS](#)] — This classification involves creating digital signatures using hashing methods. The drawback is that a signer needs to keep a track of all of the messages that have been signed, and that there is a limit to the number of signatures that can be produced. New methods, though, integrate into Merkle Trees, which allows for the signer to use the same keys to sign multiple entities.

Future Crypto

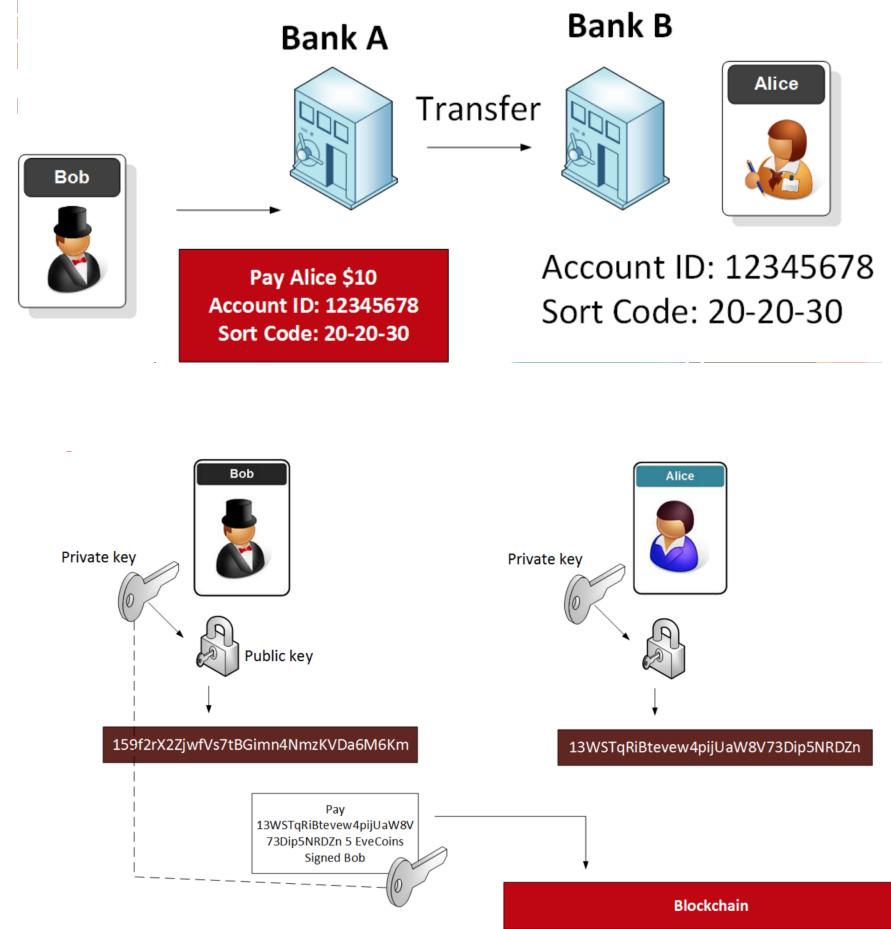
Tokenisation



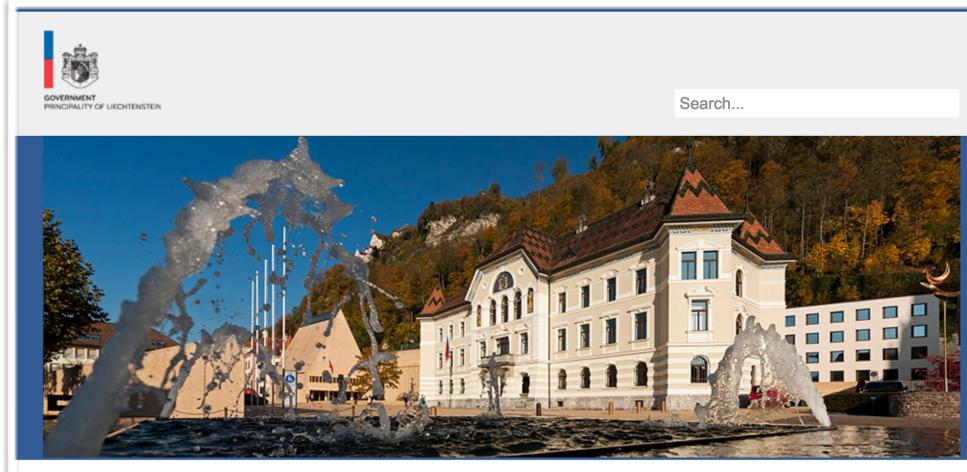
Blockchain Act



Blockchain Act



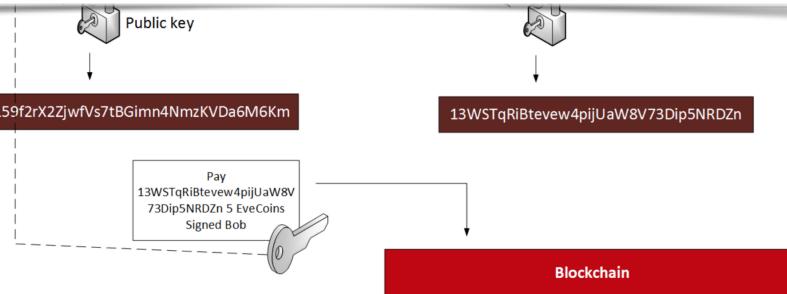
Blockchain Act



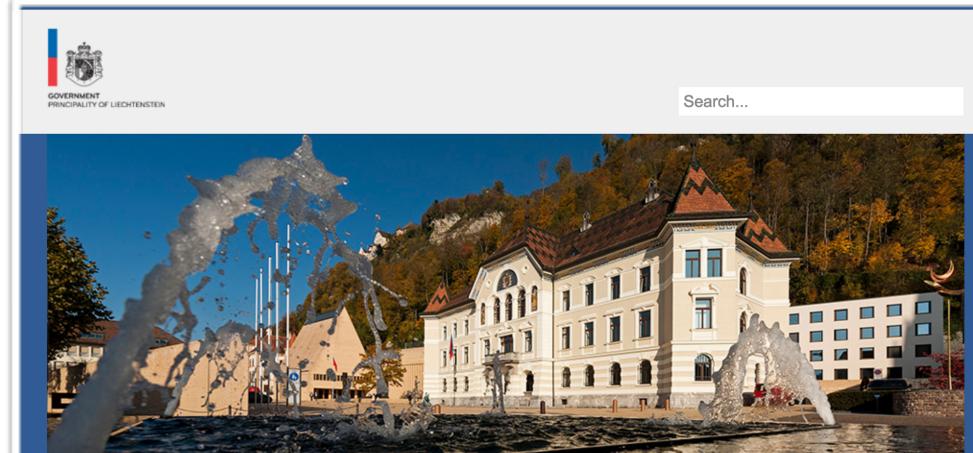
29.08.2018

Consultation launched on Blockchain Act

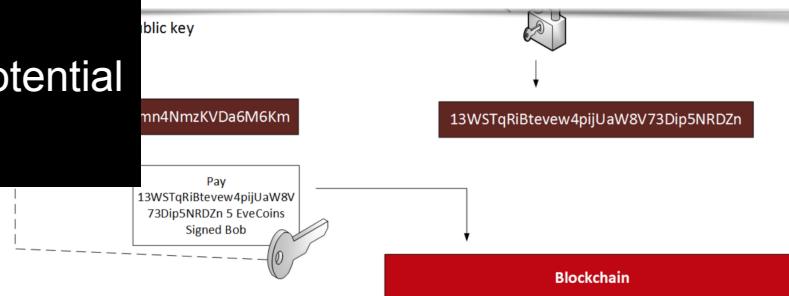
In its meeting of 28 August 2018, the Government adopted the consultation report on the Law on Transaction Systems Based on Trustworthy Technologies (TT) (Blockchain Act; TT Act; VTG) and the amendment of further laws.



Blockchain Act



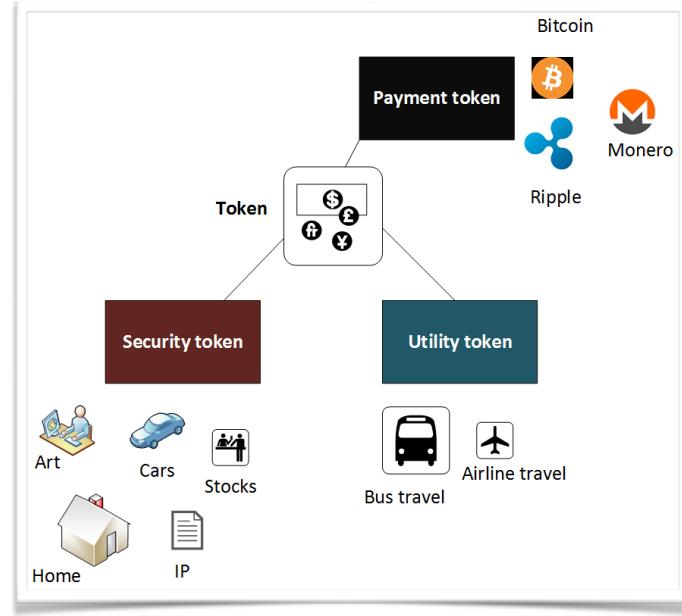
enable the transformation of the 'real' world to blockchain systems while ensuring legal certainty, thereby opening up the full application potential of the token economy.



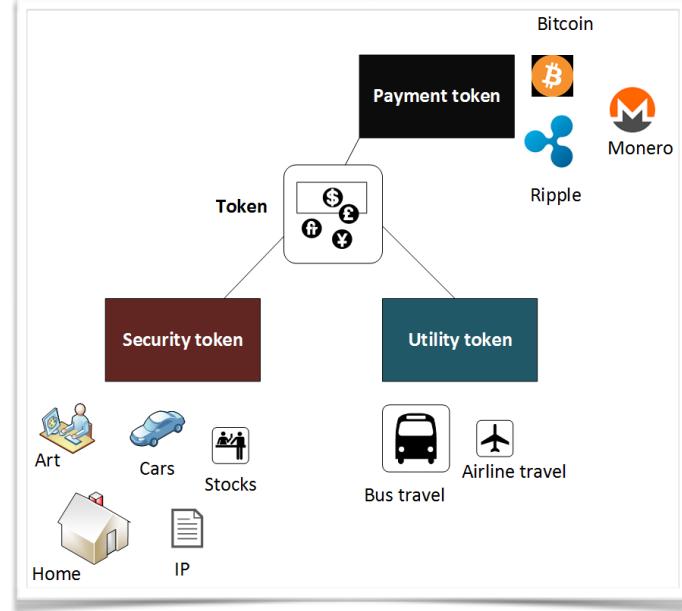
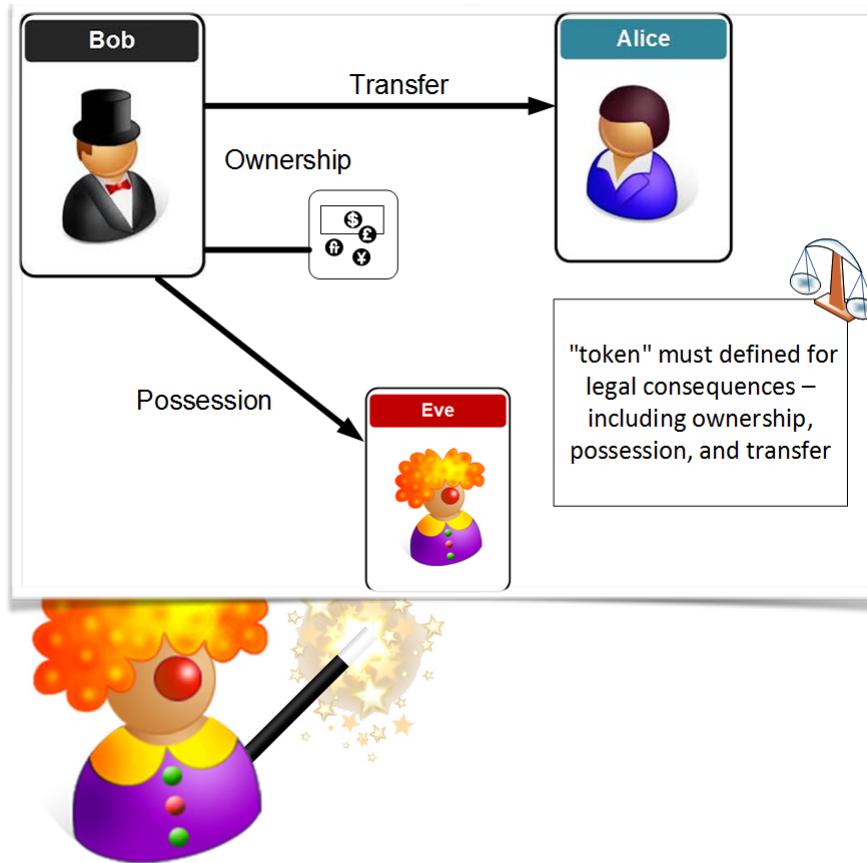
Blockchain Act



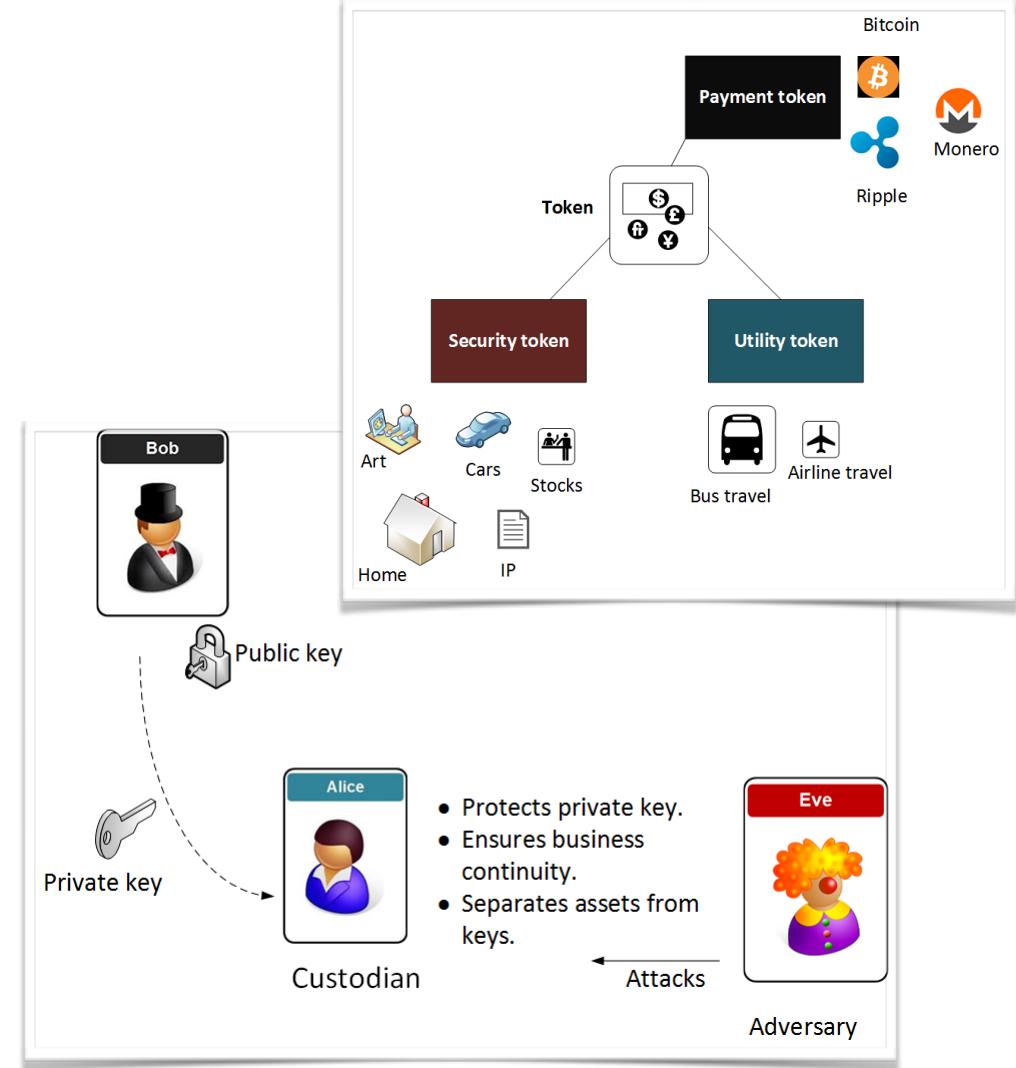
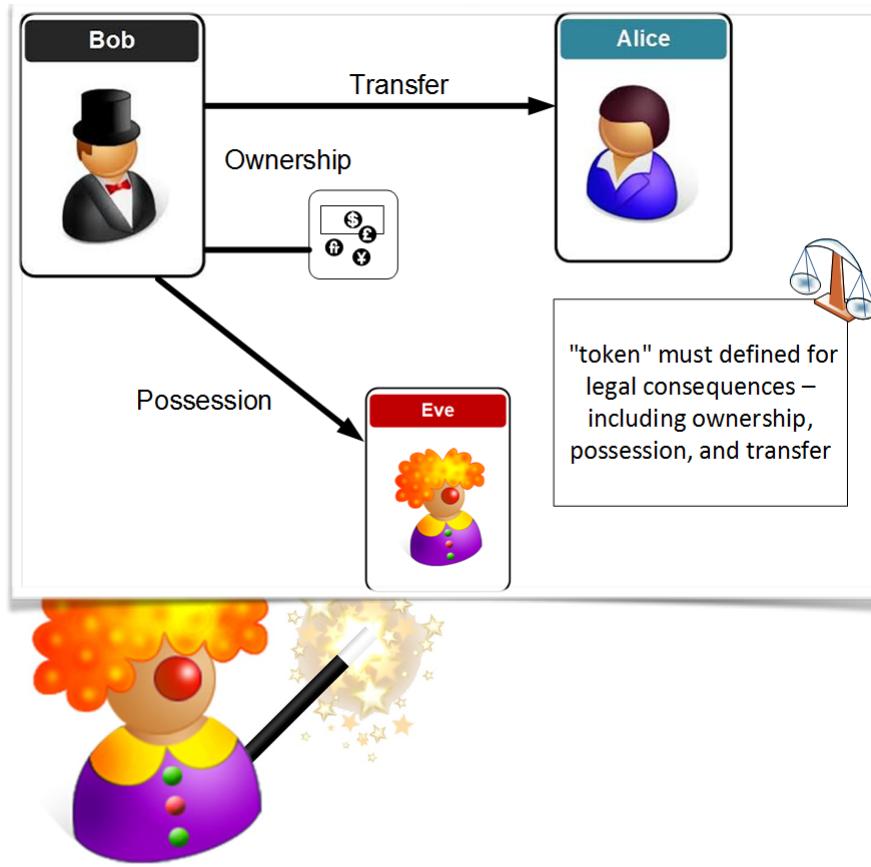
Blockchain Act



Blockchain Act



Blockchain Act



Blockchain Act

Subject and purpose (Art. 1 VE-VTG): This defines that the main focus is around the protection of users, and to thus build confidence in tokens.

Trusted technologies (Art. 3 VE-VTG): This defines the technology that is required to build a VT.

Definitions (Art. 5 VE-VTG): This defines a token as something that defines claims of a person to the rights to goods.

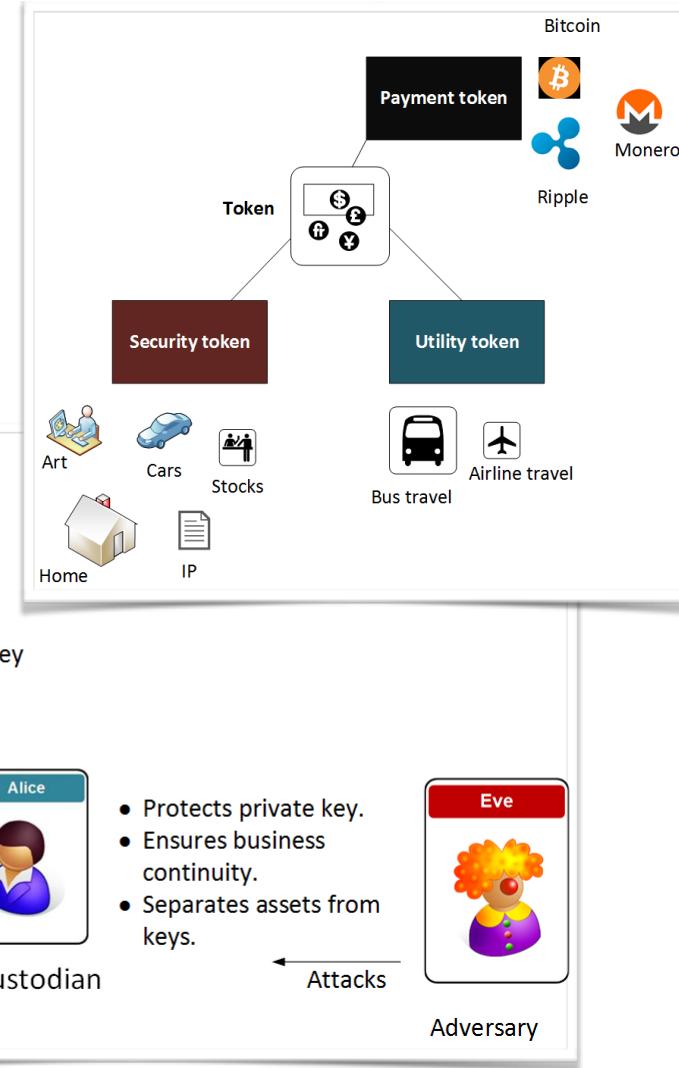
Rights of disposal (Art. 6 ff. VE-VTG): This defines the rights to transfer tokens, and is normally defined by the owner of a private key signing the transaction. A disposition is defined as the transfer of the disposition authorization on the token. Within the Act, it is defined that a buyer has the rights to dispose of a token, even if the seller was not authorized to dispose of the same token.

Requirements for VT service providers (Art. 13 ff. VE-VTG): This defines the entities who will perform services within the VT. These entities must provide an organisational structure, control mechanisms and a minimum amount of capital.

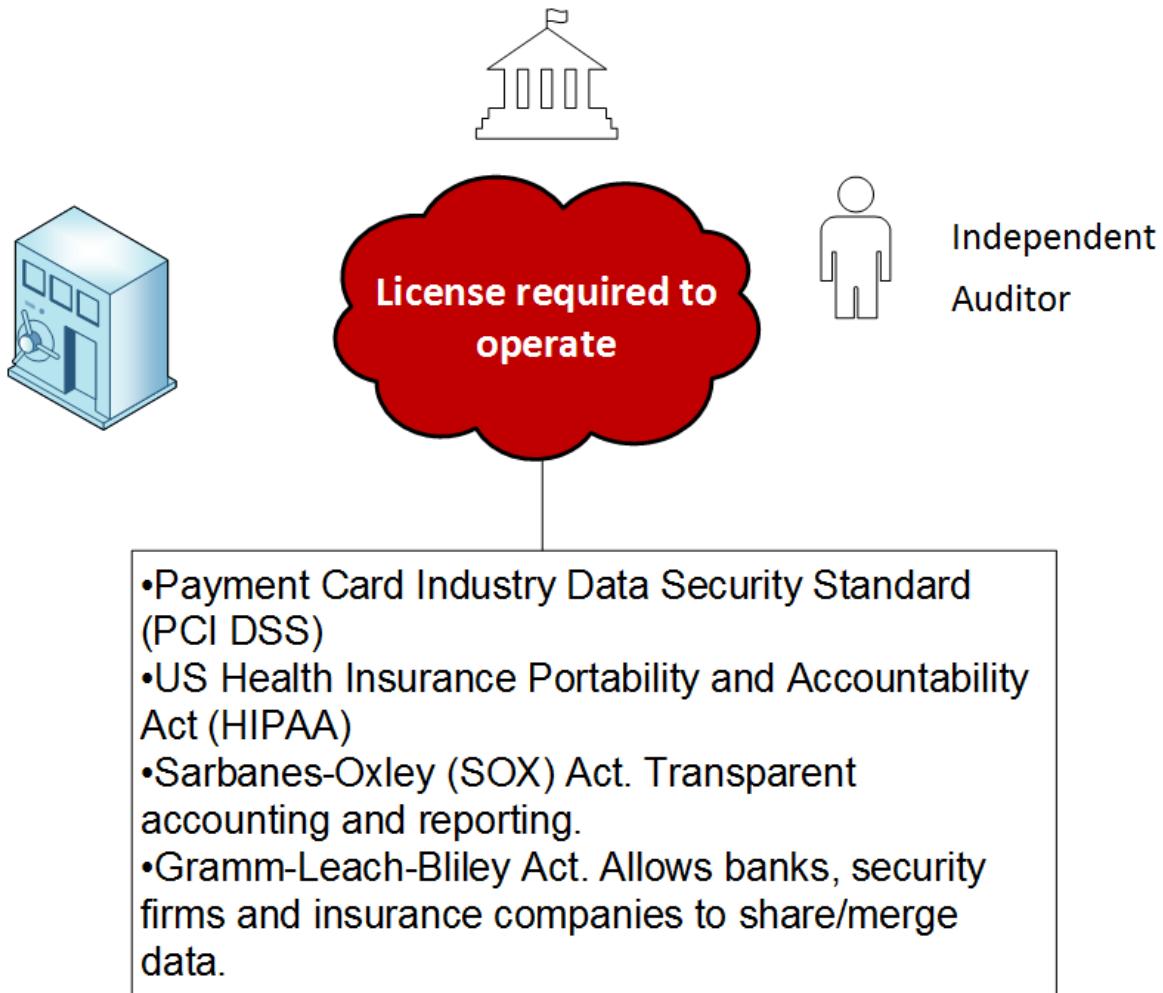
Basic information on the issuance of tokens (Art. 28 ff. VE-VTG): This defines the assurance in the issuing of tokens and their legal requirements. They must provide a minimum amount of information, such as the technology used, the purpose of the token, and any risks. There should be at least 10 years of issuance, and to also prevent token cloning, along with prevention of a token not being released with the same rights.

Obligation to register (Art. 36 ff. VE-VTG): This defines that service providers must register into the Financial Market Authority (FMA) before starting their commercial operation.

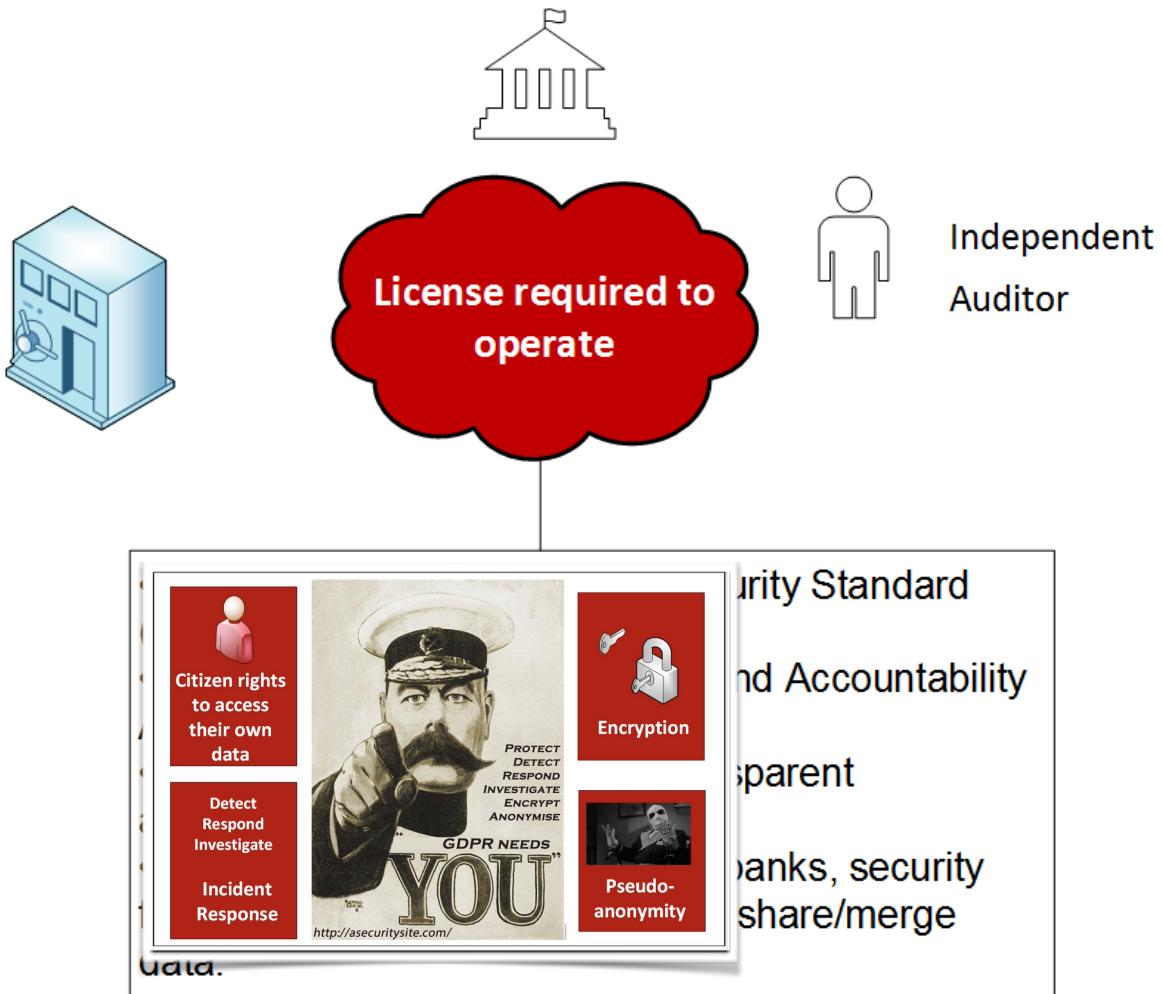
Supervision (Art. 42 ff. VE-VTG): This defines that the FMA implements the



Audit Compliance



Audit Compliance



Surrogate identifiers

Personally Identifiable Information (PII)



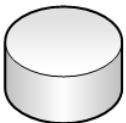
PAN – Primary
Account Number

ID=543 611 041

Name: Bobby Smith

Address: 10 Eve Row

Date of Birth: 5/5/55



Surrogate mapping
table

Real

Surrogate

ID=543 611 041

ID=741 534 011

ID=533 841 943

ID= 666 001 845

Transactions



Surrogate
Identifier

ID=741 534 011

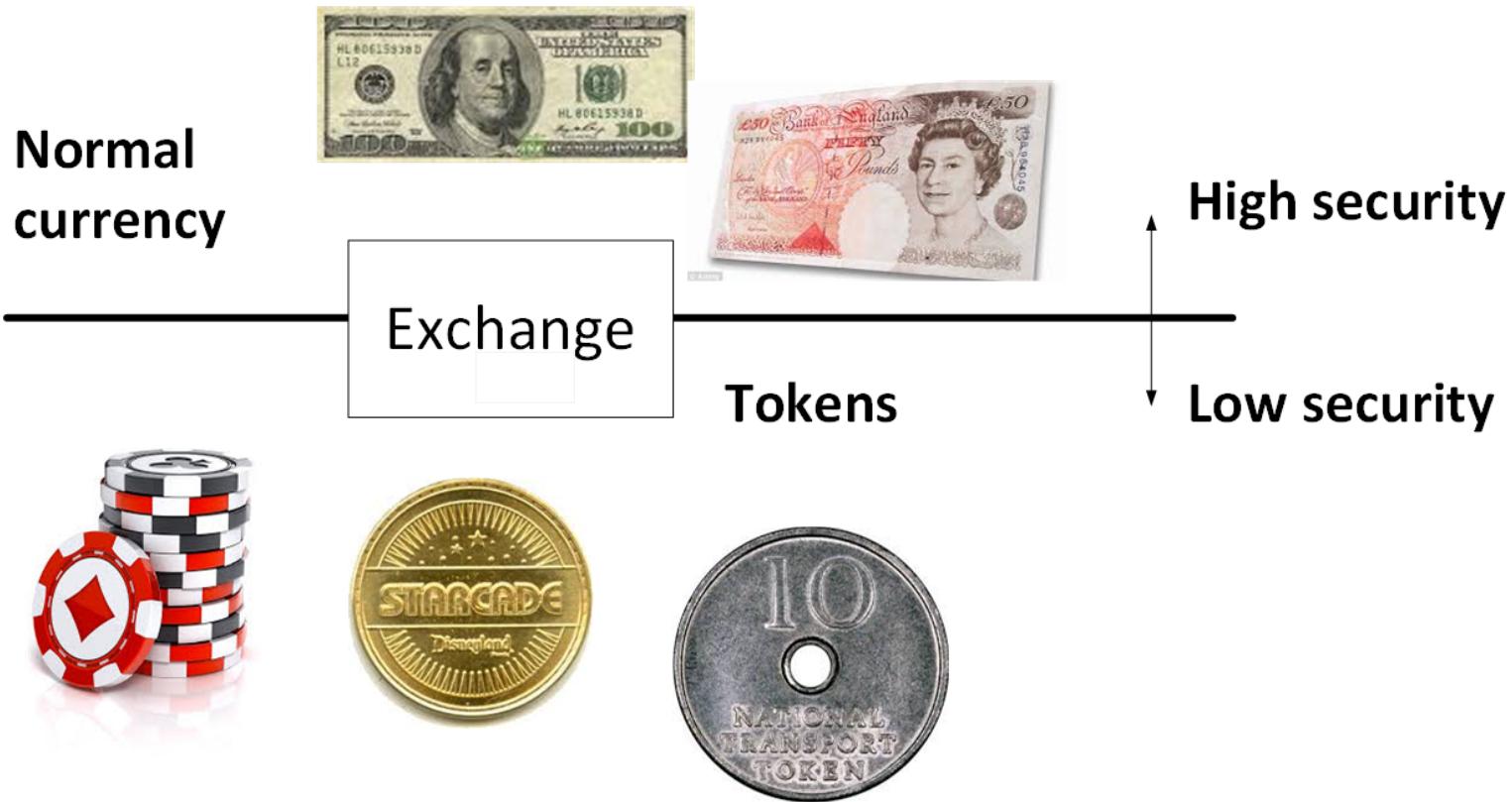
ID

741 534 001 Pay 666 001 845 \$10
532 550 423 Pay 741 534 011 \$190

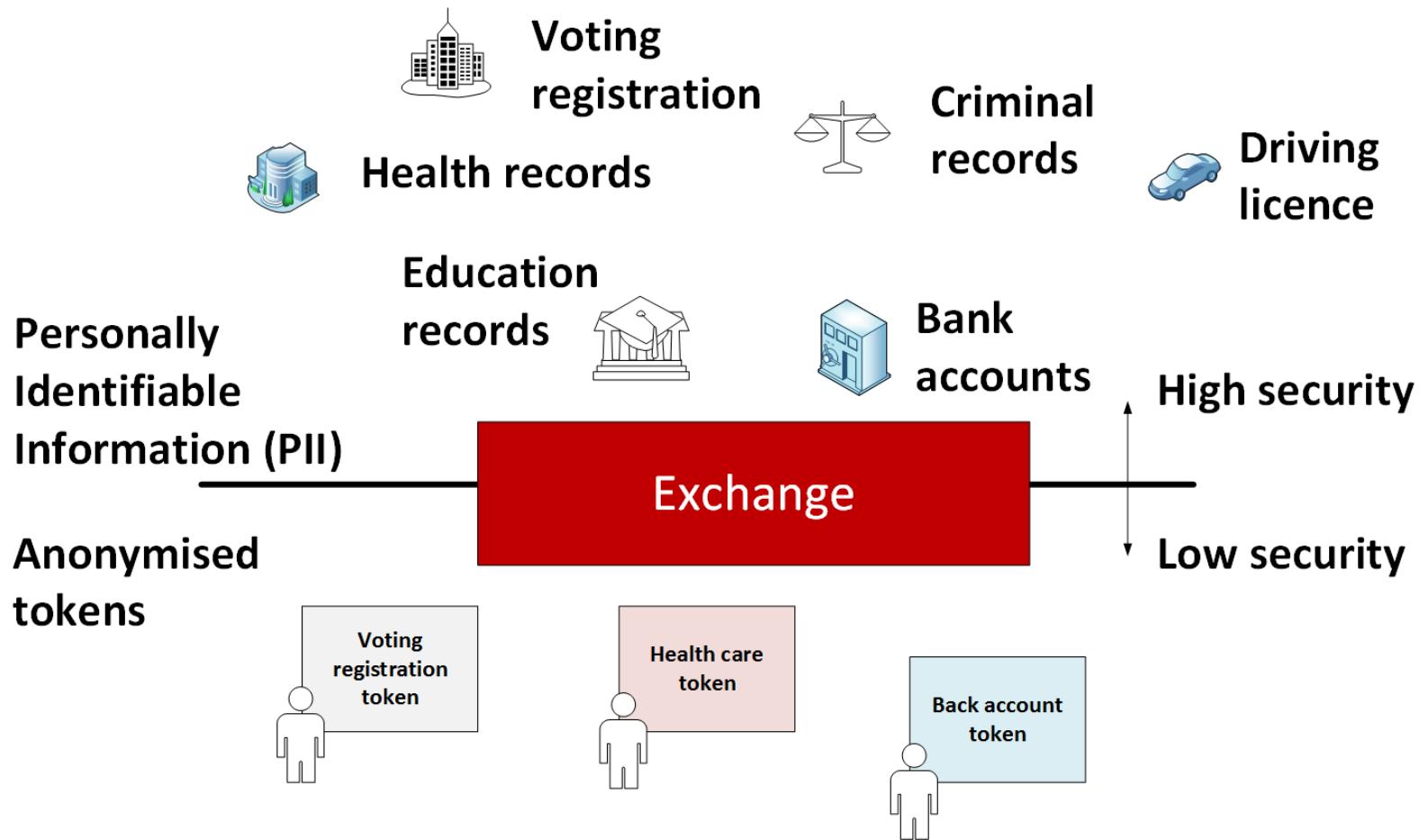
Transaction



Tokenization with currency

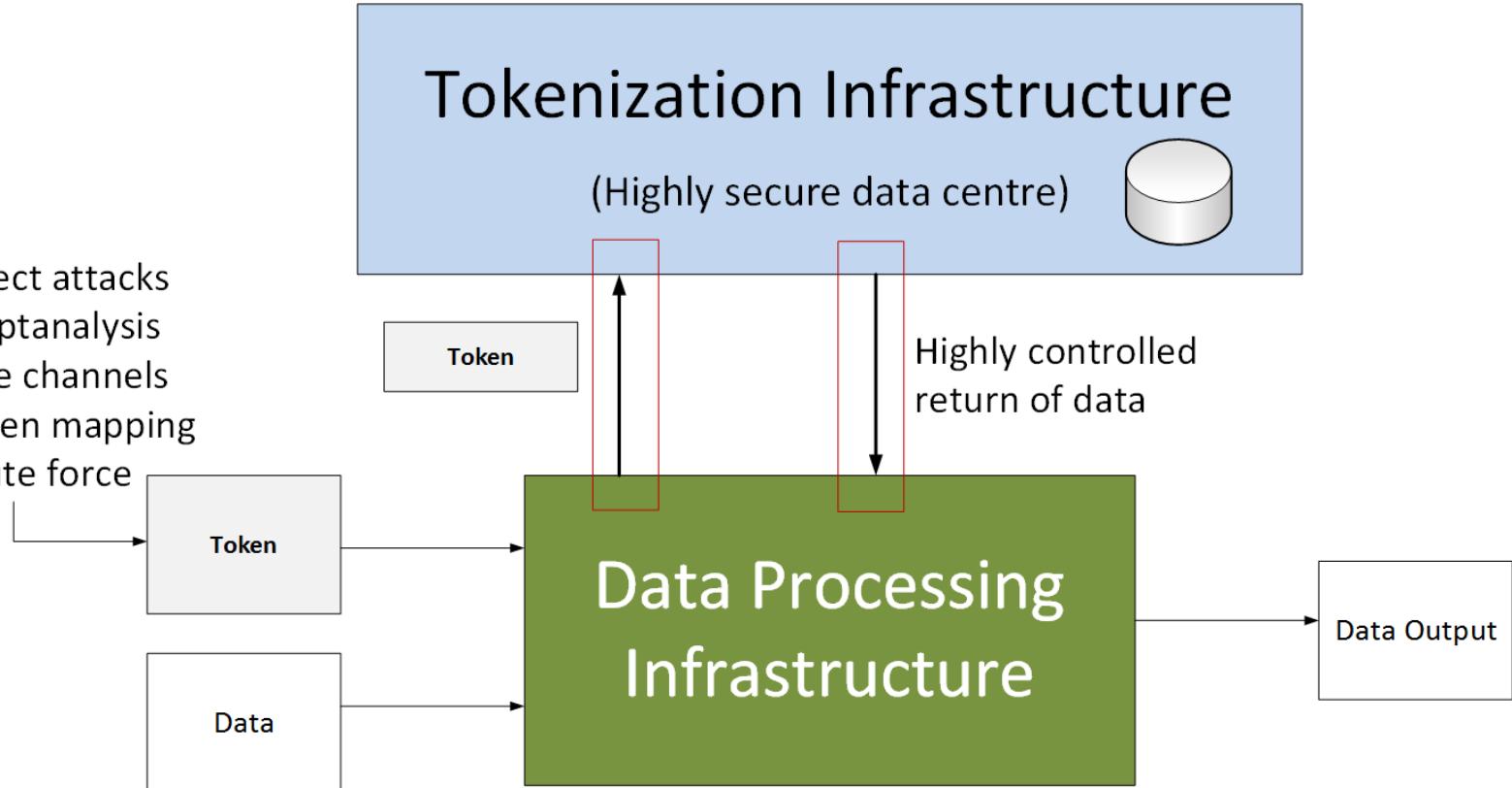


Tokenization with data

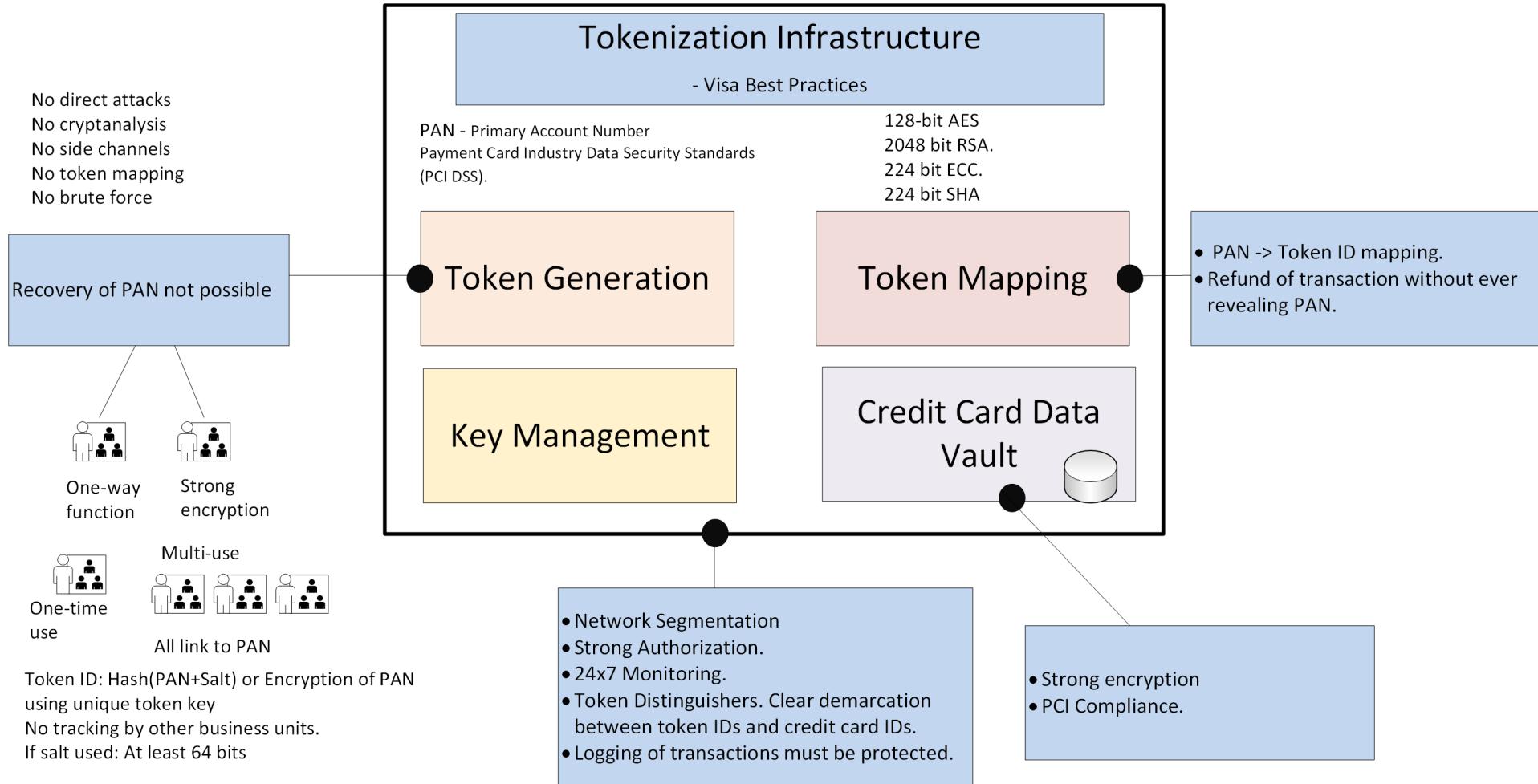


Tokenization with data

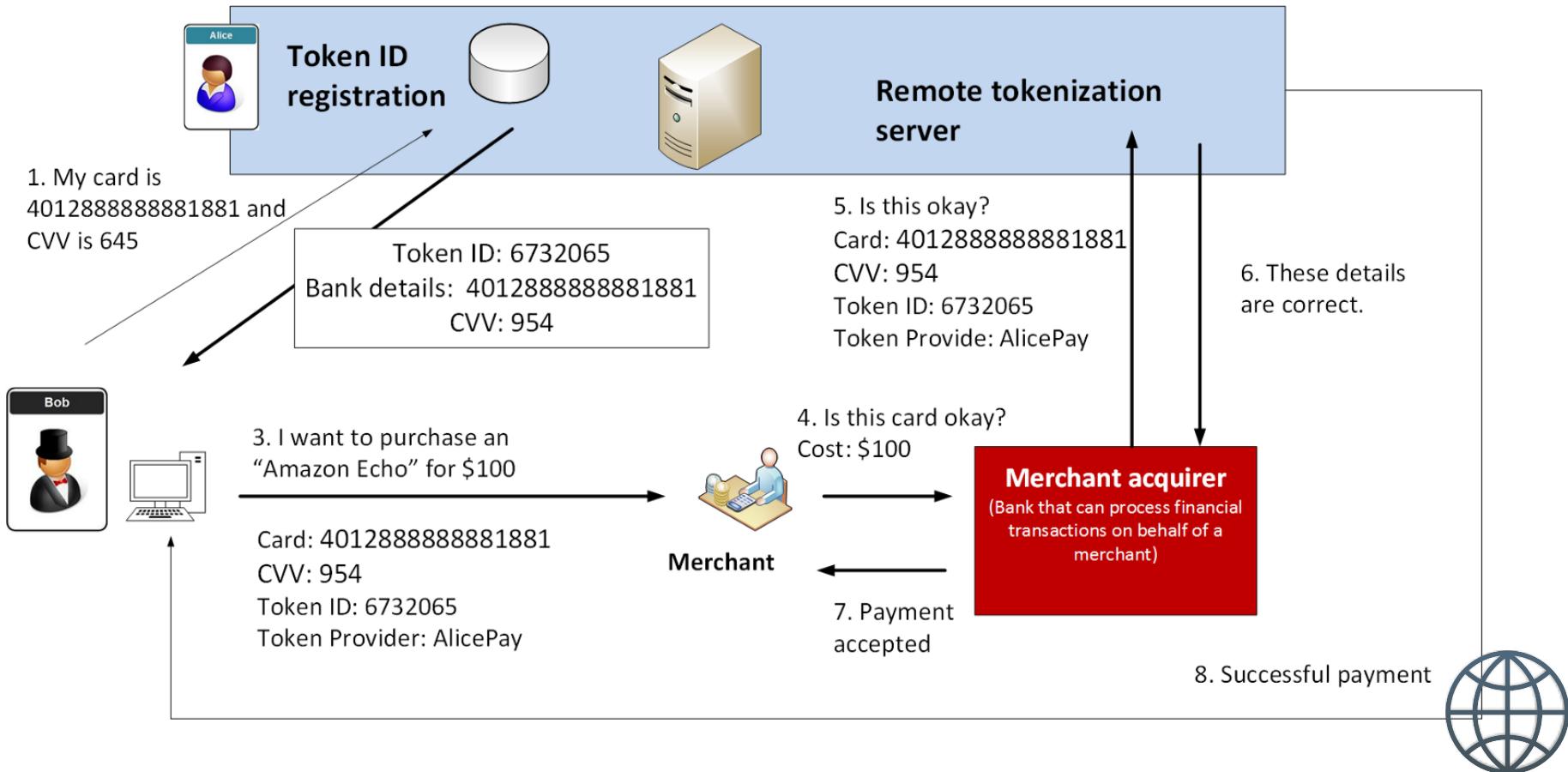
- No direct attacks
- No cryptanalysis
- No side channels
- No token mapping
- No brute force



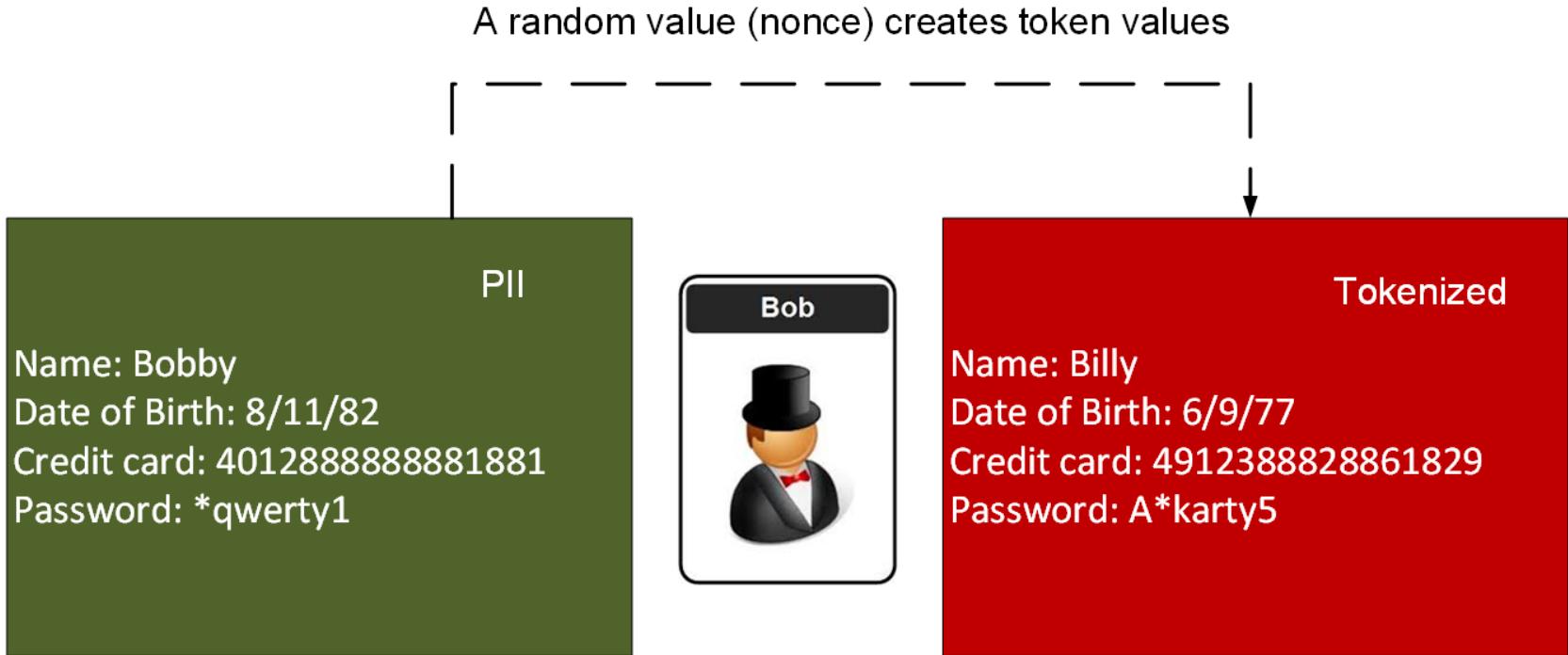
Visa Best Practice for Tokenization



Token Mapping



Token Mapping



Future Crypto

Prof Bill Buchanan OBE

<http://asecuritysite.com>

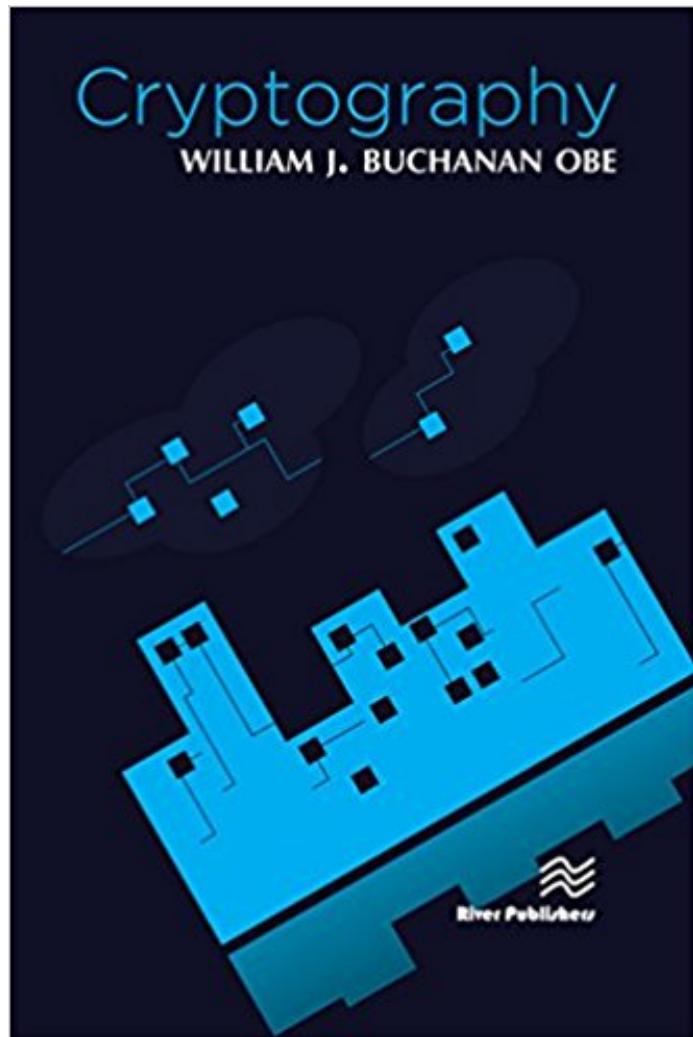
Twitter: billatnapier

Zero Knowledge Proofs and Credential Signing



BLOCKPASS
IDENTITY
LAB

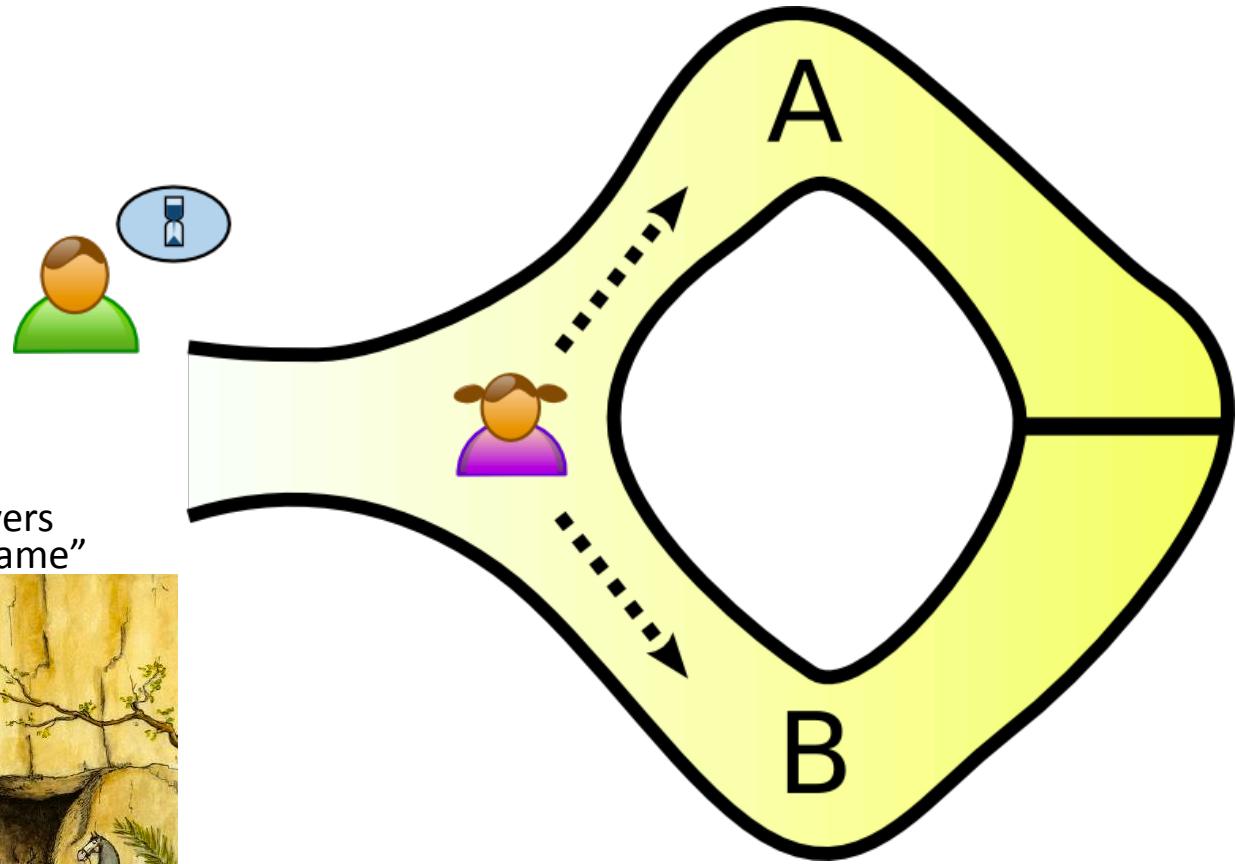
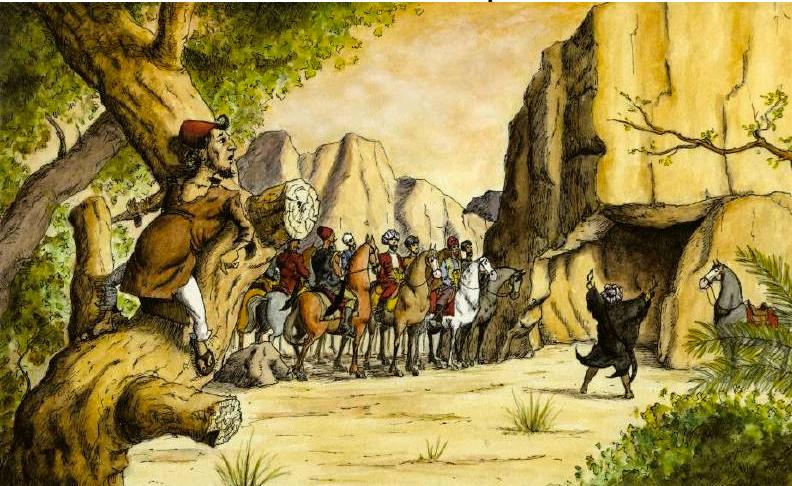
World-leading Collaboration between
Blockpass IDN and Edinburgh Napier University



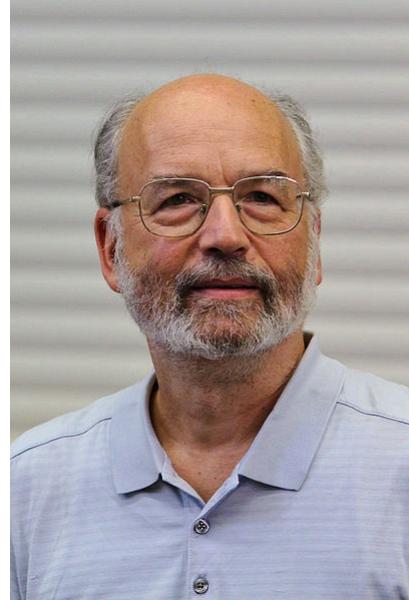
Zero-knowledge Proof

- Peggy is the prover.
- Victor is the verifier.

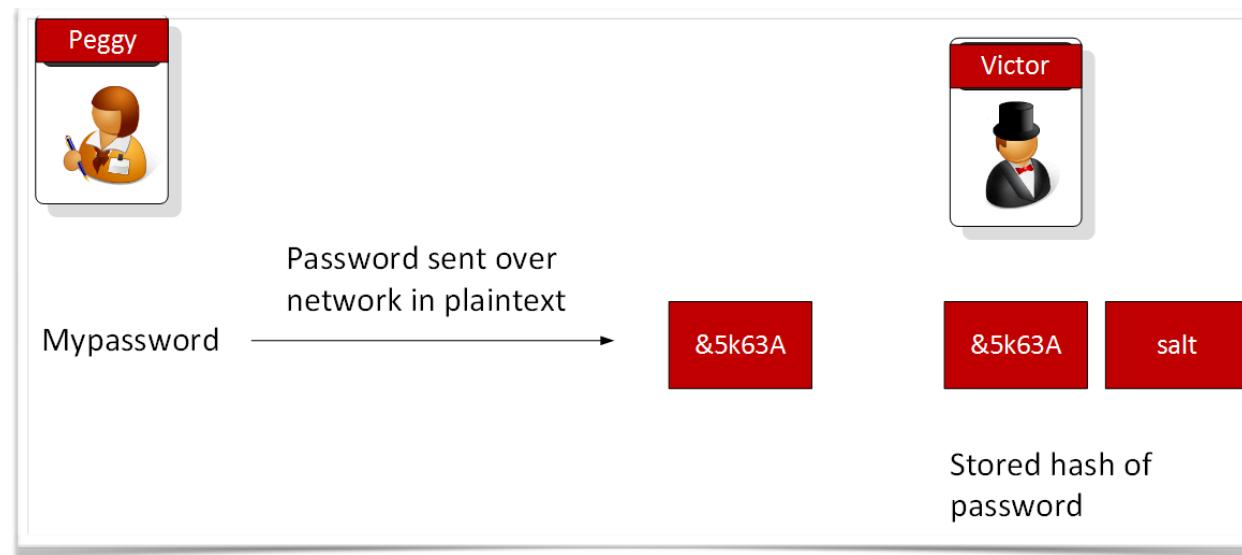
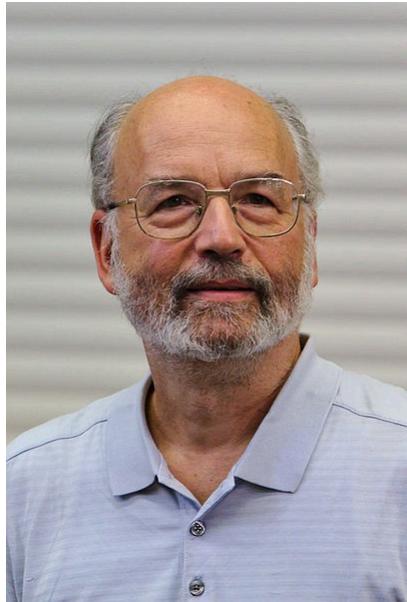
Ali Baba - poor woodcutter - discovers the secret of a thieves as "open sesame"



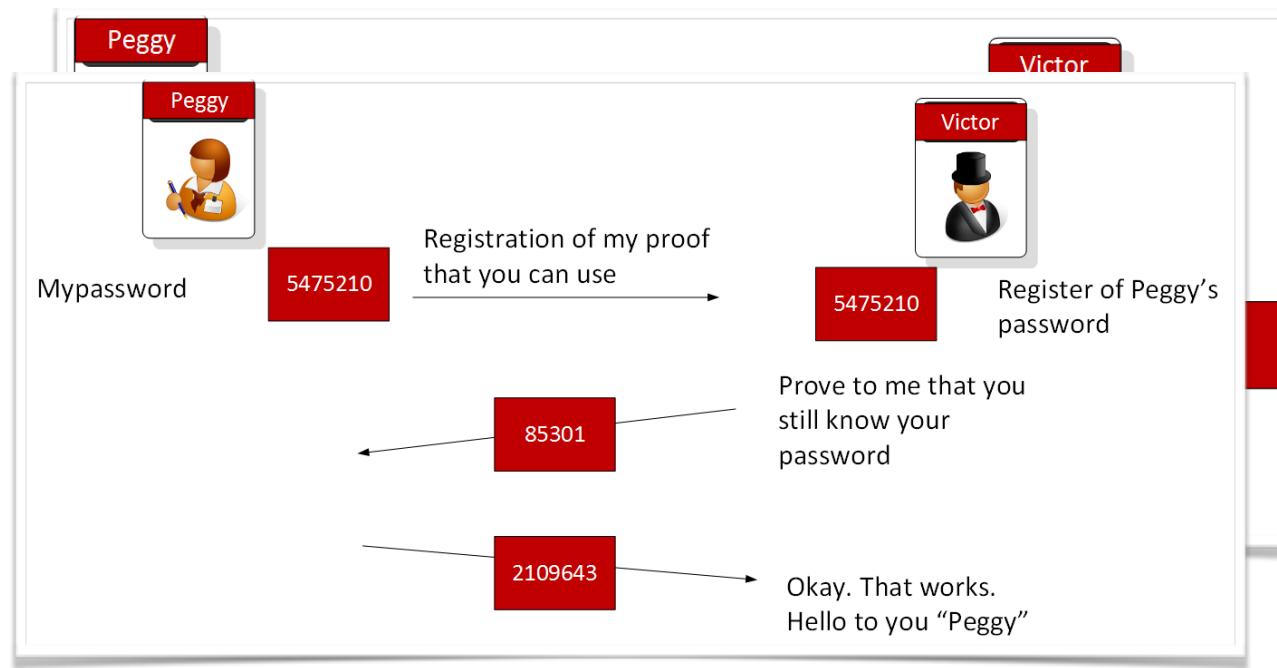
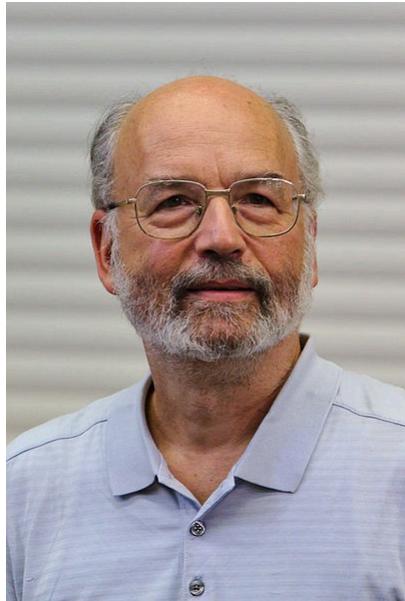
Zero Knowledge Proof: Fiat-Shamir



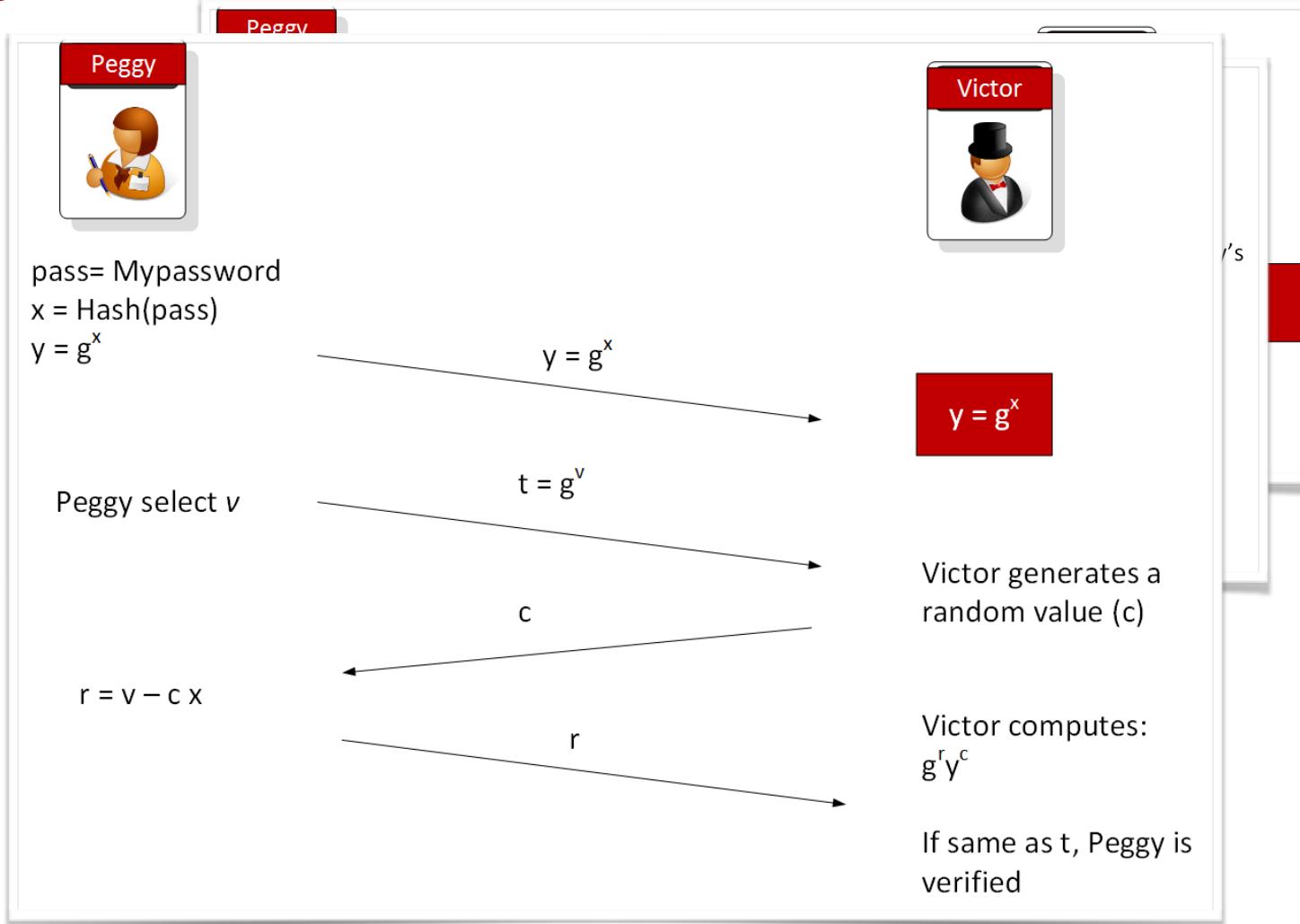
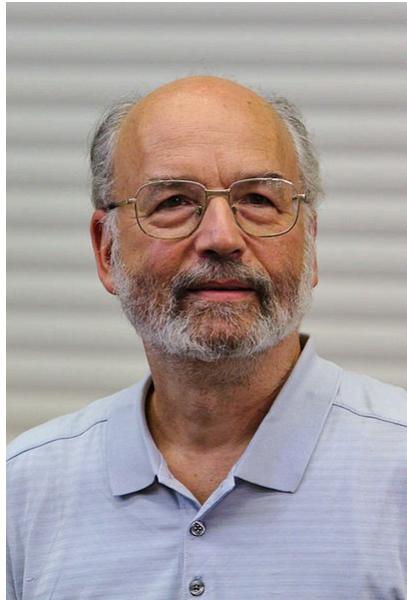
Zero Knowledge Proof: Fiat-Shamir



Zero Knowledge Proof: Fiat-Shamir

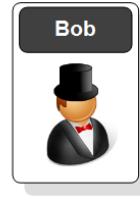


Zero Knowledge Proof: Fiat-Shamir



Bob proves and Alice verifies - Non-interactive random oracle access

Bob is the prover and knows the value of x



Bob and Alice agree on G and p

Alice is the verifier

Shared Secret: $y = G^x$

Random value: v

Commitment: $t = g^v$

Challenge: $c = \text{Hash}(g, y, t)$

Response: $r = v - cx \pmod p$

Prove to me you still know x !

$$\begin{aligned} g^r y^c &= g^{v-cx} y^c \\ &= g^{v-cx} (g^x)^c \\ &= g^{v-cx+cv} \\ &= t \end{aligned}$$

Check $t' = g^r y^c$
Recheck $c = H(g, y, t')$

(r, c)

Future Crypto

Prof Bill Buchanan OBE

<http://asecuritysite.com>

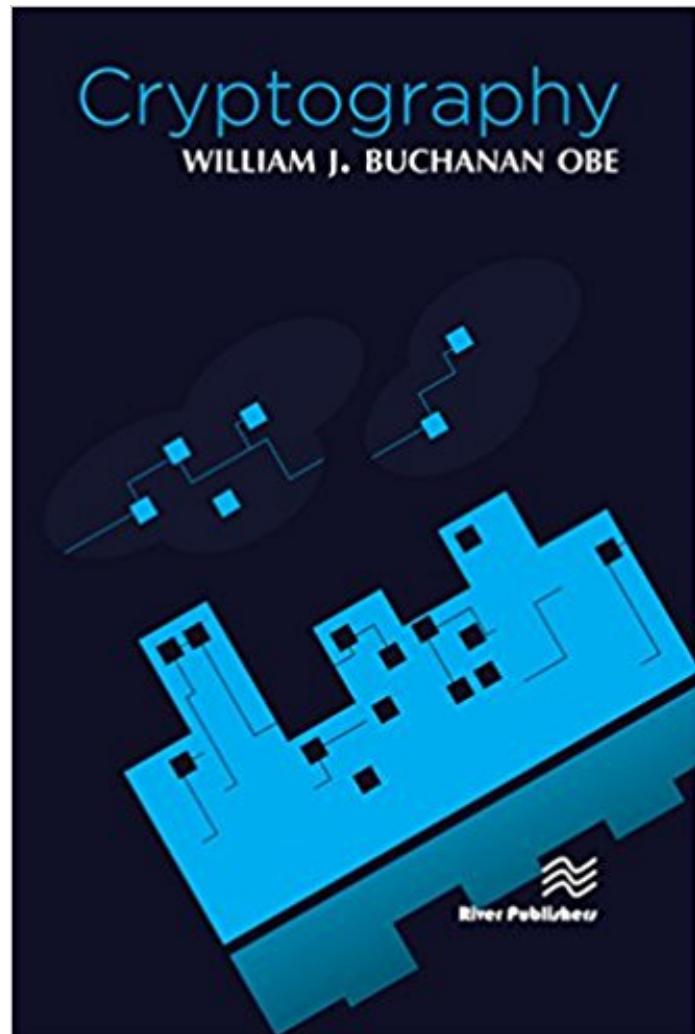
Twitter: billatnapier

Homomorphic Encryption

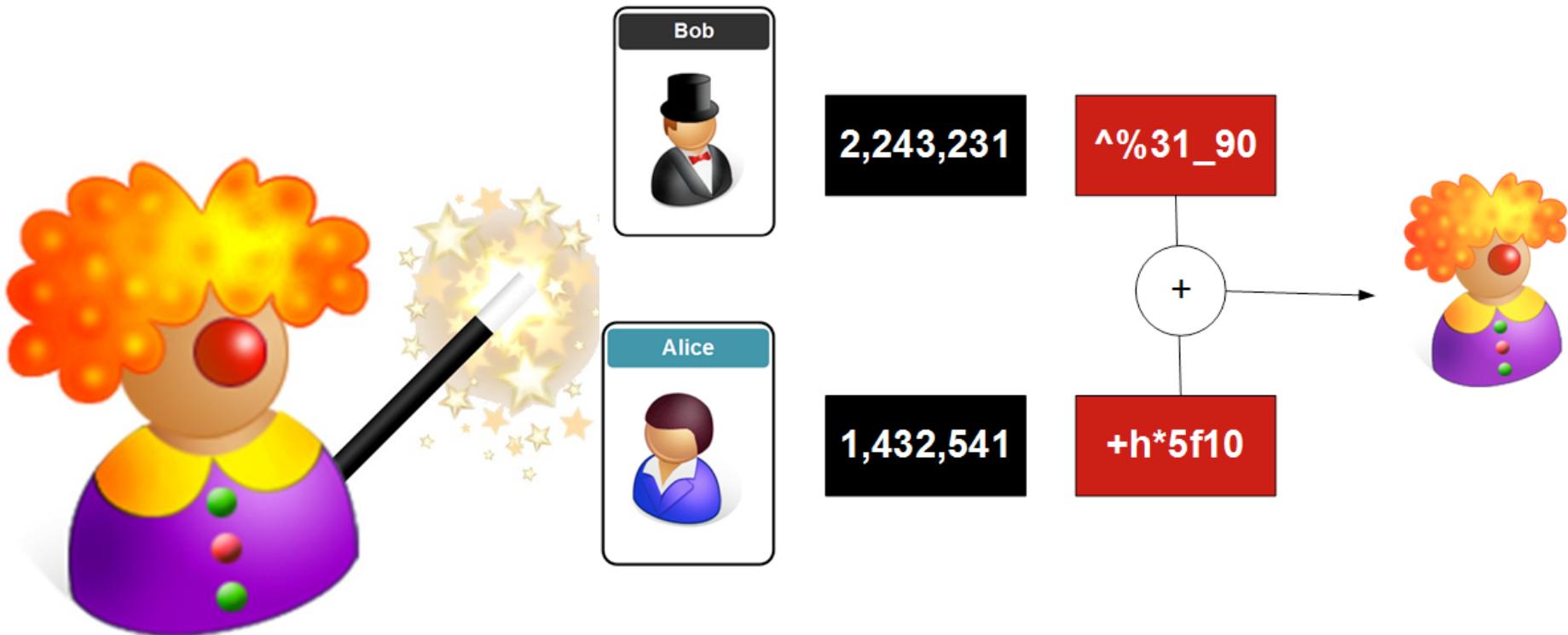


BLOCKPASS
IDENTITY
LAB

World-leading Collaboration between
Blockpass IDN and Edinburgh Napier University



Homomorphic Encryption



Homomorphic Encryption



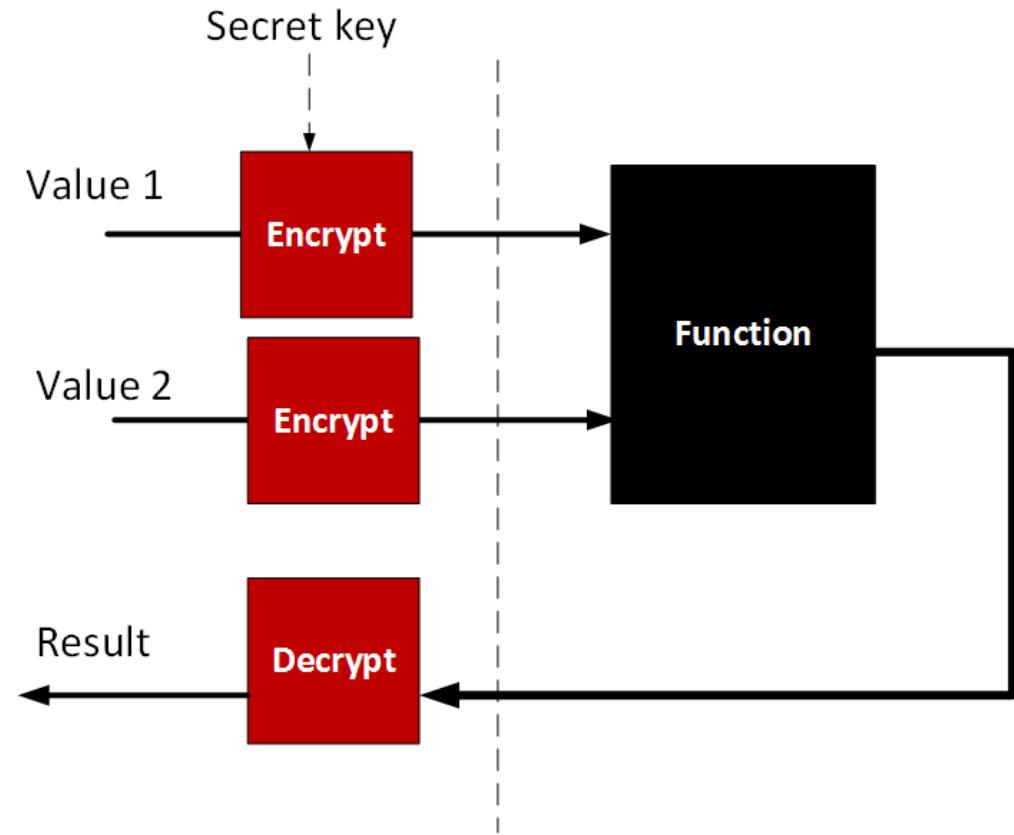
How do we find out who is older? Without revealing age

If Bob has worked for a number of days and Alice has also worked for a number of days. How to calculate their income without revealing days worked?

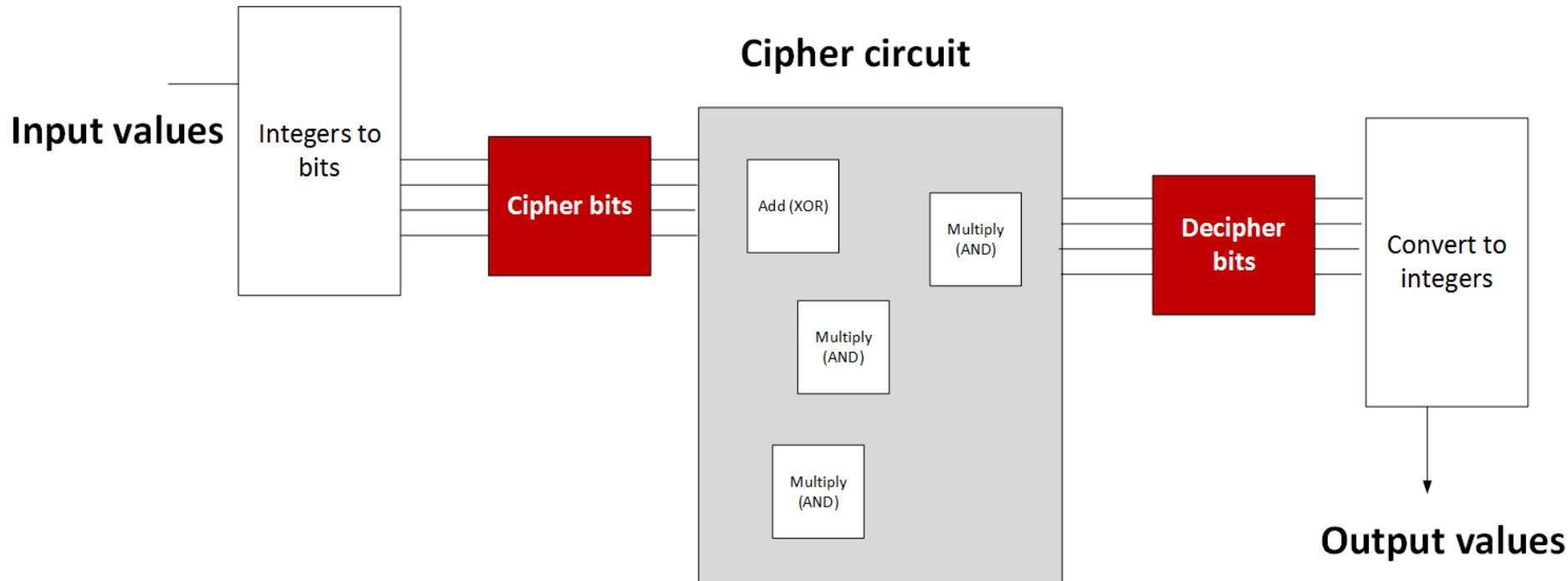
What is their total income? Without revealing their income

The diagram illustrates three scenarios of homomorphic encryption using icons of Bob and Alice. In each scenario, Bob's icon is on the left and Alice's icon is on the right, both enclosed in grey rounded rectangles. The top scenario asks how to determine who is older without revealing age. The middle scenario asks how to calculate total income given the number of days worked by Bob and Alice without revealing the individual days. The bottom scenario asks what their total income is without revealing individual incomes. The icons show Bob in a tuxedo and Alice in a business suit, with Alice holding a pen and paper.

Full Homomorphic Encryption - DGKV (Dijk, Gentry, Halevi and Vaikuntanathan)



Full Homomorphic Encryption - DGKV (Dijk, Gentry, Halevi and Vaikuntanathan)



Full Homomorphic Encryption - DGKV (Dijk, Gentry, Halevi and Vaikuntanathan)



$$c = p \times q + 2 \times r + m$$

$$d = (c \mod p) \mod 2$$

Future Crypto

Prof Bill Buchanan OBE

<http://asecuritysite.com>

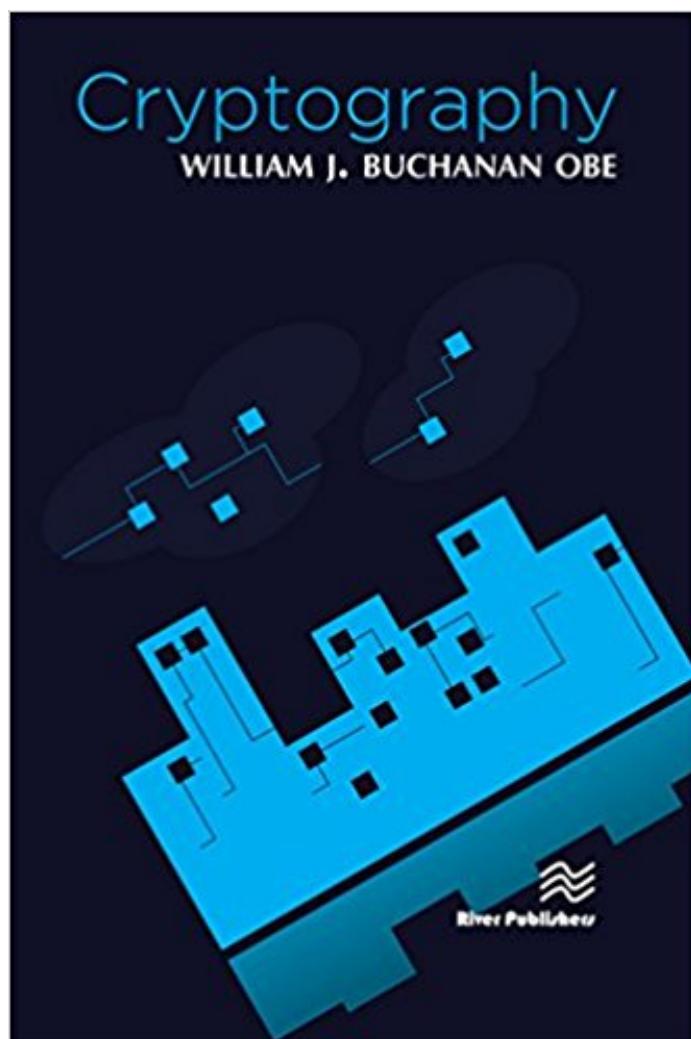
Twitter: billatnapier

zkSnarks



BLOCKPASS
IDENTITY
LAB

World-leading Collaboration between
Blockpass IDN and Edinburgh Napier University



zCash



Zcash uses ZKP

Humans that were at fault with a new Zcoin hack (Zcash) and which involved making transactions of £561,000 (\$699,000), and with an associated profit of £349,000 (\$435,000).

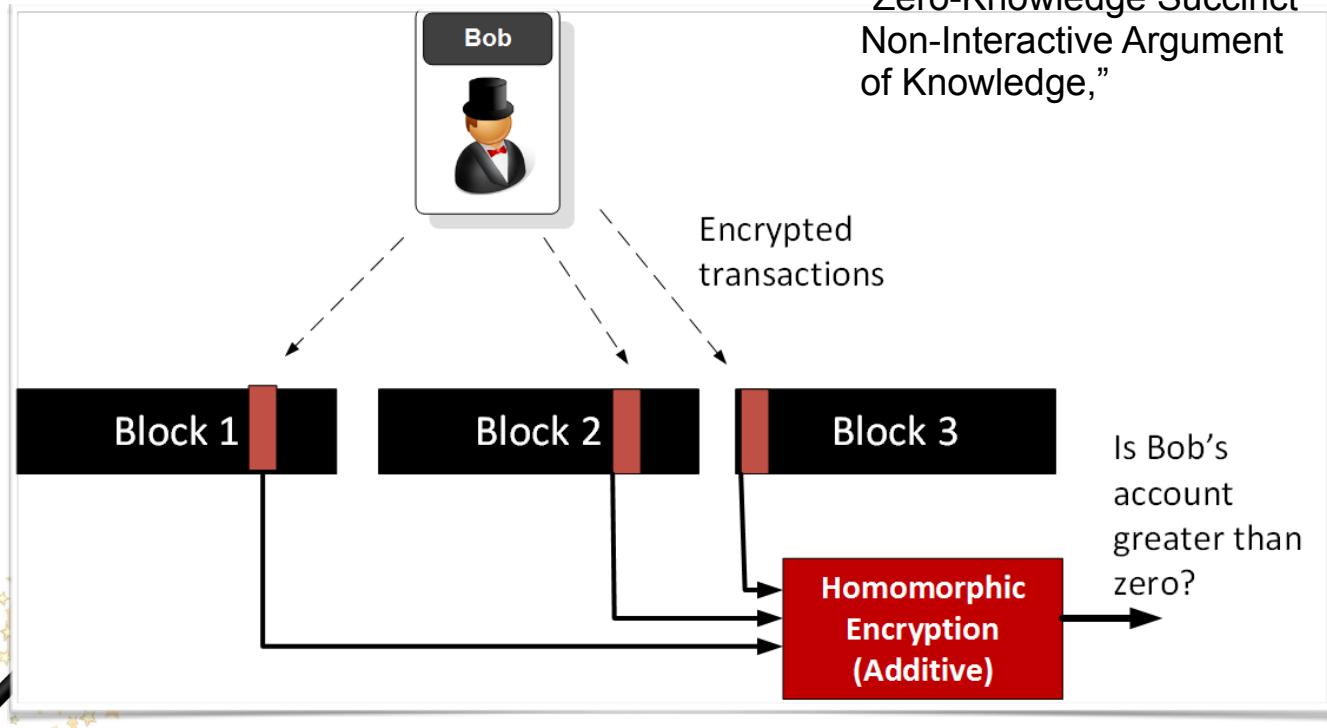
It happened with a single additional character in the source code, and allowed the creation of a Zerocoin spend transaction, and which was able to extract 370,000 Zcoins.

zk-SNARK stands for “Zero-Knowledge Succinct Non-Interactive Argument of Knowledge,”

Benchmark	Time in seconds	Min	Max	Change	Trend
time createjoinsplit	44.64126	44.61028	44.98179	-0.65	-4.48%
time parameterloading	2.64027	2.63953	2.70091	-0.16	-2.60%
time solveequihash	30.78956	14.24525	107.95399	-32.47	-19.73%
time solveequihash 2 threads	49.32094	15.41534	115.41706	-0.46	-3.35%
time validatelargetx	0.60553	0.60509	0.60841	-0.06	0.10%
time verifyequihash	0.00155	0.00154	0.00371	-0.58	0.09%
time verifyjoinsplit	0.02776	0.02775	0.05627	-1.38	-4.41%
Average				-5.11	-4.91%

Zero-knowledge Proofs

zk-SNARK stands for
“Zero-Knowledge Succinct
Non-Interactive Argument
of Knowledge,”



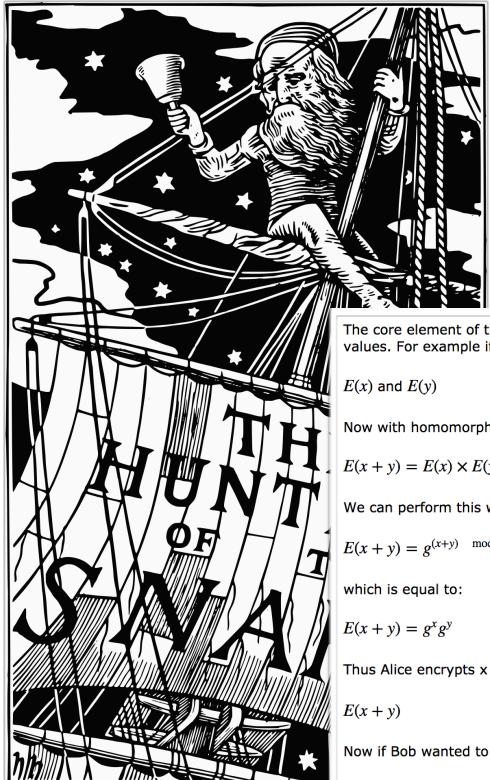


With HH we aim to prove that Alice knows two values (x, y) to equal a given answer (ans)

Let's say you want to prove that Alice can produce two numbers which add up to 8. Overcomes the major problem of Blockchain - which is the lack of privacy in computations and transactions.

Blind Evaluation

Bob doesn't want Alice to know the values that he is using and Alice doesn't want Bob to know the method she is using to compute a result

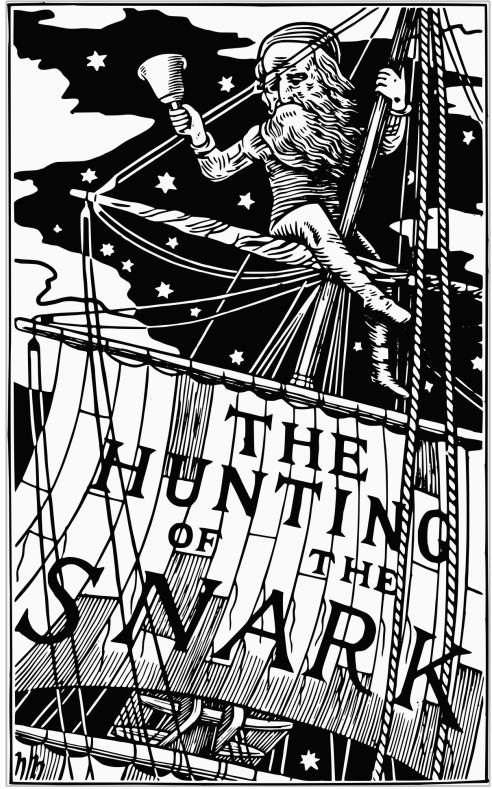


With HH we aim to prove that Alice knows two values (x, y) to equal a given answer (ans)

Let's say you want to prove that Alice can produce two number which adds up to 8. Overcomes the major problem of Blockchain - which is the lack of privacy in

Blind Evaluation

ant Alice to known t he is using and vant Bob to know e is using to ult



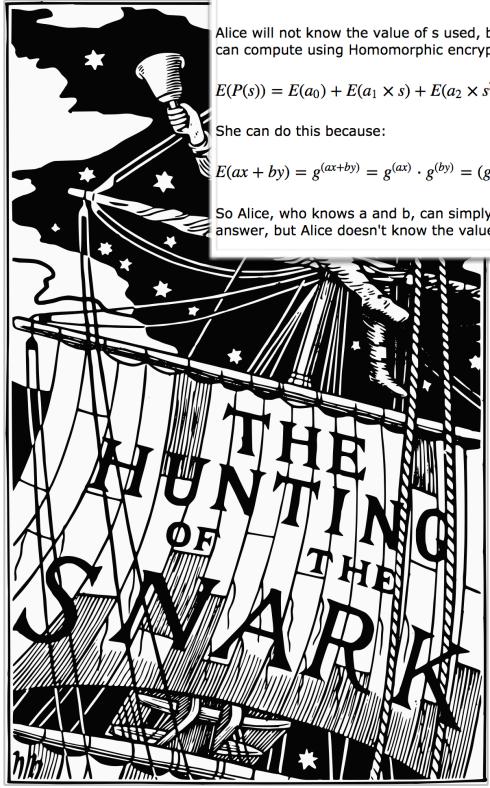
With HH we aim to prove that Alice knows two values (x, y) to equal a given answer (ans)

Let's say you want to prove that Alice can produce two numbers which add up to 8. Overcomes the major problem of Blockchain - which is the lack of privacy in computations and transactions.

Blind Evaluation

Bob doesn't want Alice to know the values that he is using and Alice doesn't want Bob to know the method she is using to compute a result

zkSNARK



For this we use polynomials, and where we have an equation such as:

$$P(x) = a_0 + a_1x + a_2x^2$$

Bob sends all the elements - known as hidings - of the computation for a value of s:

$$E(a_0), E(a_1 \times s) \text{ and } E(a_2 \times s^2)$$

Alice will not know the value of s used, but she knows the "wiring" of the function. In this case she knows that it is a simple adding operation, so she can compute using Homomorphic encryption:

$$E(P(s)) = E(a_0) + E(a_1 \times s) + E(a_2 \times s^2)$$

She can do this because:

$$E(ax + by) = g^{(ax+by)} = g^{(ax)} \cdot g^{(by)} = (g^x)^a \cdot (g^y)^b = E(x)^a \cdot E(y)^b$$

So Alice, who knows a and b, can simply raise the values received to the polynomial factors and multiply and return to Bob. Bob then knows the answer, but Alice doesn't know the value of s used.

Hidden Homomorphic

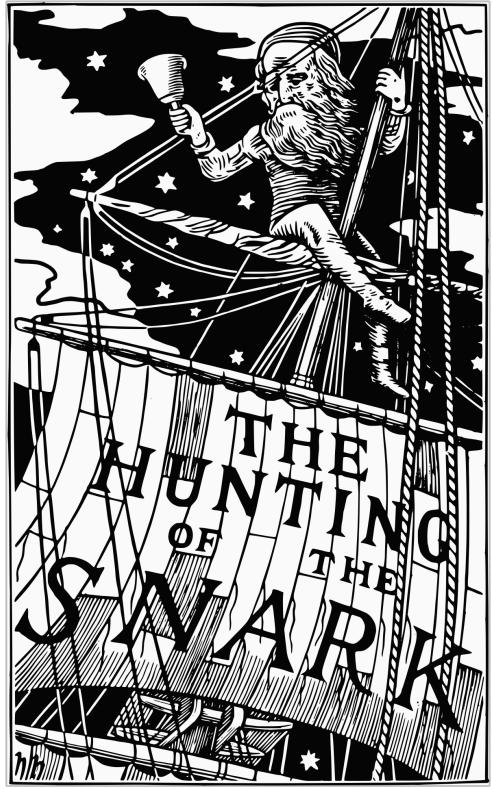
we aim to prove that Alice
no values (x, y) to equal a
answer (ans)

that
per
mes the

major problem of Blockchain -
which is the lack of privacy in
computations and transactions.

Blind Evaluation

Bob doesn't want Alice to know
the values that he is using and
Alice doesn't want Bob to know
the method she is using to
compute a result



With HH we aim to prove that Alice knows two values (x, y) to equal a given answer (ans)

Let's say you want to prove that Alice can produce two numbers which add up to 8. Overcomes the major problem of Blockchain - which is the lack of privacy in computations and transactions.

Blind Evaluation

Bob doesn't want Alice to know the values that he is using and Alice doesn't want Bob to know the method she is using to compute a result

Future Crypto

Prof Bill Buchanan OBE

<http://asecuritysite.com>

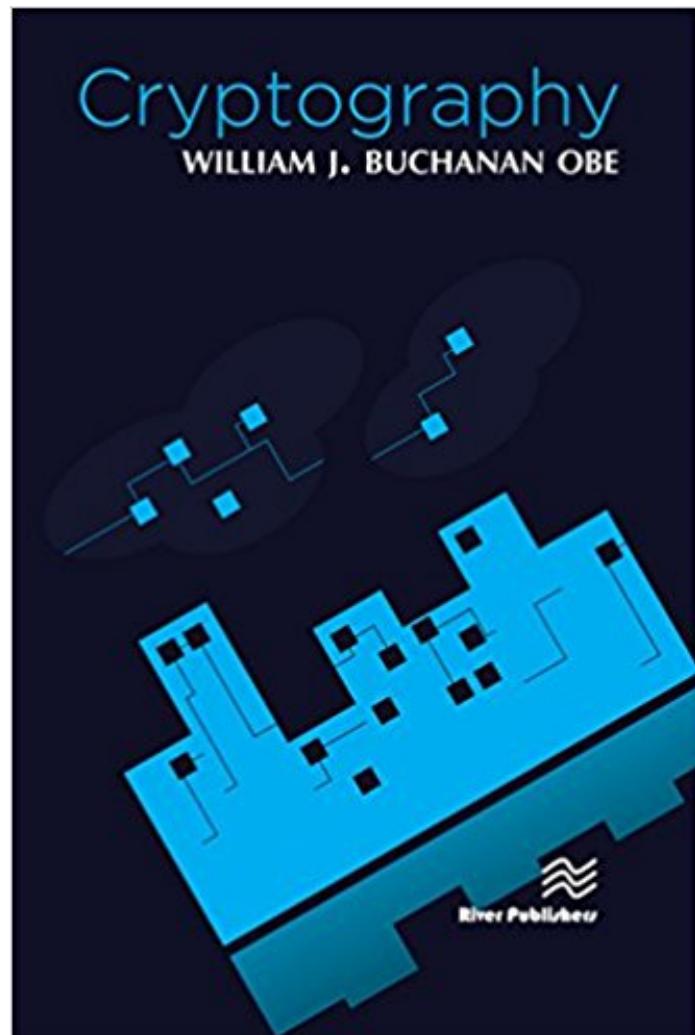
Twitter: billatnapier

Range Proofs, Pedersen
Commitments, etc

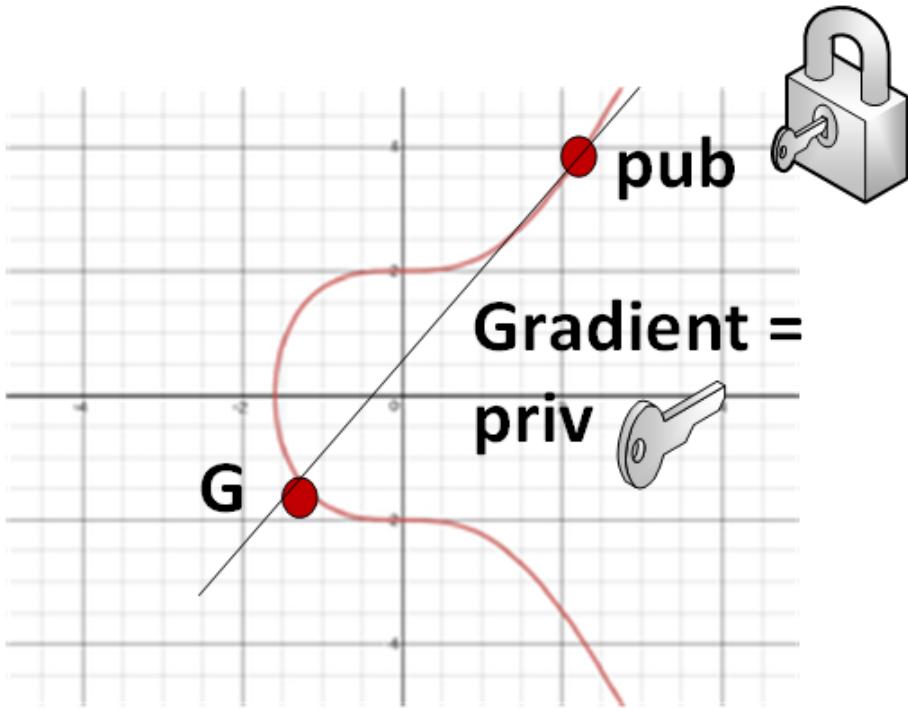


BLOCKPASS
IDENTITY
LAB

World-leading Collaboration between
Blockpass IDN and Edinburgh Napier University



Public and private keys with ECC



Private key:

0xc9f4f55bdeb5ba0bd337f2dbc952a5439e20ef
9af6203d25d014e7102d86aaeeL

Public key:

0xc44370819cb3b7b57b2aa7edf550a9a5410c23
4d27aff497458bbbfc8b6a327,
0x52a1a3e222cd89cbd2764b69bd9b0ea5c4fd6c
a28861e1f2140eff9c2e76487

G:

(50662630222773436695787188951685343262
50603453777594175500187360389116729240L
,
32670510020758816978083085130507043184
47127338065924327593890433575733748242
4L)

Pedersen Commitment and Mimblewimble



Pedersen Commitment and Mimblewimble

Adding a blinding value

For this, we can obscure the values in a transaction by adding a **blinding value**. Let's say that we have a transaction value of v , we can now define this as a point on the elliptic curve (H) as:

$$v \times H$$

If we have three transactions (v_1 , v_2 and v_3), we can create a sum total of:

$$\text{Total} = v_1 \times H + v_2 \times H + v_3 \times H = (v_1 + v_2 + v_3) \times H$$

We can thus determine the sum of the transactions. But we could eventually find-out the values of v_1 , v_2 and v_3 , as they would always appear as the same value when multiplied by H . We can now add a blinding factor by adding a second point on another elliptic curve (G) and a private key (r). A transaction value is then (as defined as a Pedersen Commitment):

$$C = v \times H + r \times G$$



Pedersen Commitment and Mimblewimble

Addition a blinding value

MIMBLEWIMBLE

Tom Elvis Jedusor

19 July, 2016

****/

Introduction

/****\

Bitcoin is the first widely used financial system for which all the necessary data to validate the system status can be cryptographically verified by anyone. However, it accomplishes this feat by storing all transactions in a public database called "the blockchain" and someone who genuinely wishes to check this state must download the whole thing and basically replay each transaction, check each one as they go. Meanwhile, most of these transactions have not affected the actual final state (they create outputs that are destroyed a transaction later).

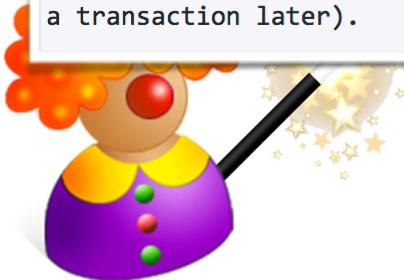
By adding a **blinding** value, we can now define this

create a sum total of:

But we could eventually always appear as the same blinding factor by adding a

second point on another elliptic curve (G) and a private key (r). A transaction value is then (as defined as a Pedersen Commitment):

$$C = v \times H + r \times G$$



Pedersen Commitment and Mimblewimble

Addition a blinding value

MIMBLEWIMBLE

Tom E.

19 Jul

Intro

/****

Bitco

data

Howeve

database

this s

check

affec

a tra



Ignotus Peverell

ignopeverell

Block or report user

Sign in to view email

by adding a blinding

Overview

Repositories 5

Stars 2

Followers 135

Following 0

Popular repositories

grin

Forked from [mimblewimble/grin](#)

Minimal implementation of the MimbleWimble protocol.

● Rust ★ 1

site-test

Test repository for a slightly more elaborate Grin Jekyll site

● HTML ★ 1 ⚡ 2

hacker

Forked from [pages-themes/hacker](#)

Hacker is a Jekyll theme for GitHub Pages

● CSS ⚡ 1

enumset

Forked from [Lymia/enumset](#)

A library for defining enums that can be used in compact bit sets.

● Rust

value is then (as defined as a Pedersen commitment).

$$C = v \times H + r \times G$$

Pedersen Commitment and Mimblewimble

Addition of a blinding value

MIMBLEWIMBLE

Tom E...
19 Ju...

Intro...

Bitco...

data ...

Howev...

databa...

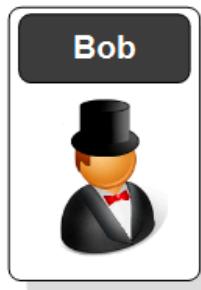
this s...

check...

affect...

a tra...

r=10
(Blinding factor)



H – Point on Elliptic Curve
G – Point on Elliptic Curve

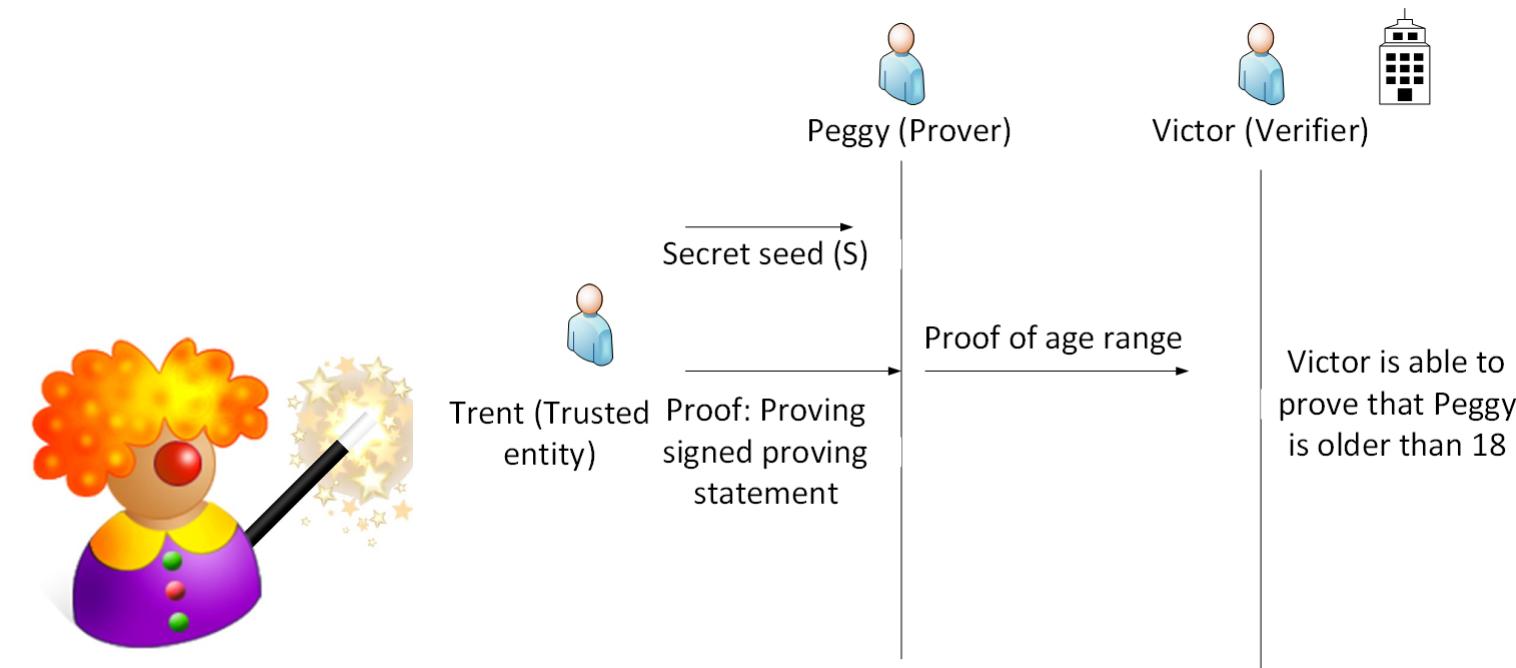
v=4 BTC



X = vH + rG

C = v×H+r×G

Range Proof



Range Proof

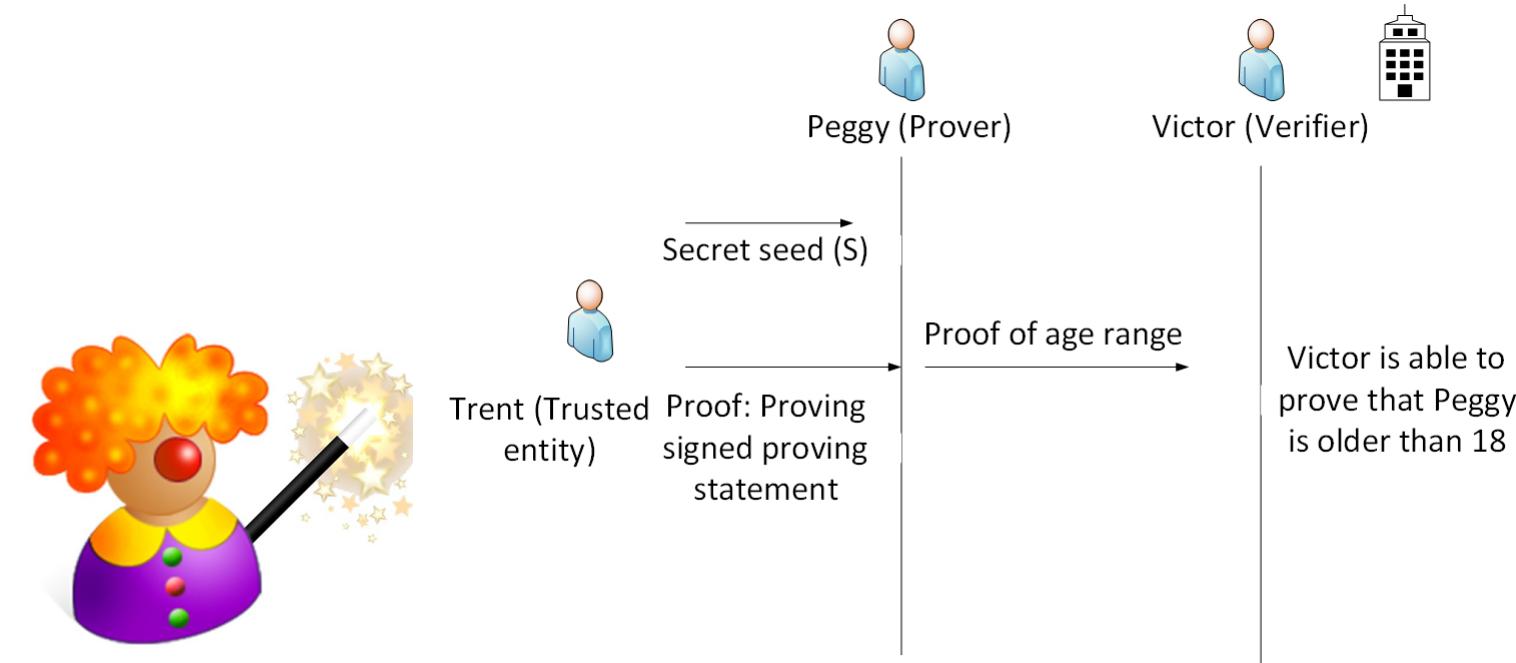
Bulletproofs: Short Proofs for Confidential Transactions and More

Benedikt Bünz^{*1}, Jonathan Bootle^{†2}, Dan Boneh^{‡1},
Andrew Poelstra^{§3}, Pieter Wuille^{¶3}, and Greg Maxwell^{||}

¹Stanford University

²University College London

³Blockstream



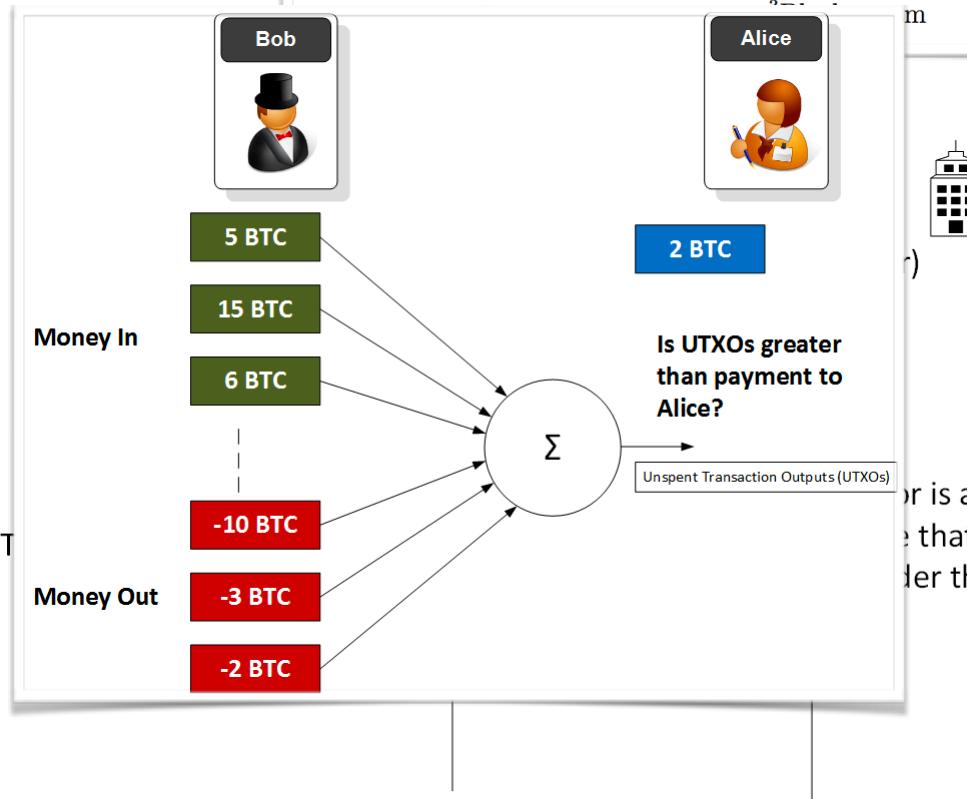
Range Proof

Bulletproofs: Short Proofs for Confidential Transactions and More

Benedikt Bünz^{*1}, Jonathan Bootle^{†2}, Dan Boneh^{‡1},
Andrew Poelstra^{§3}, Pieter Wuille^{¶3}, and Greg Maxwell^{||}

¹Stanford University

²University College London



Range Proof

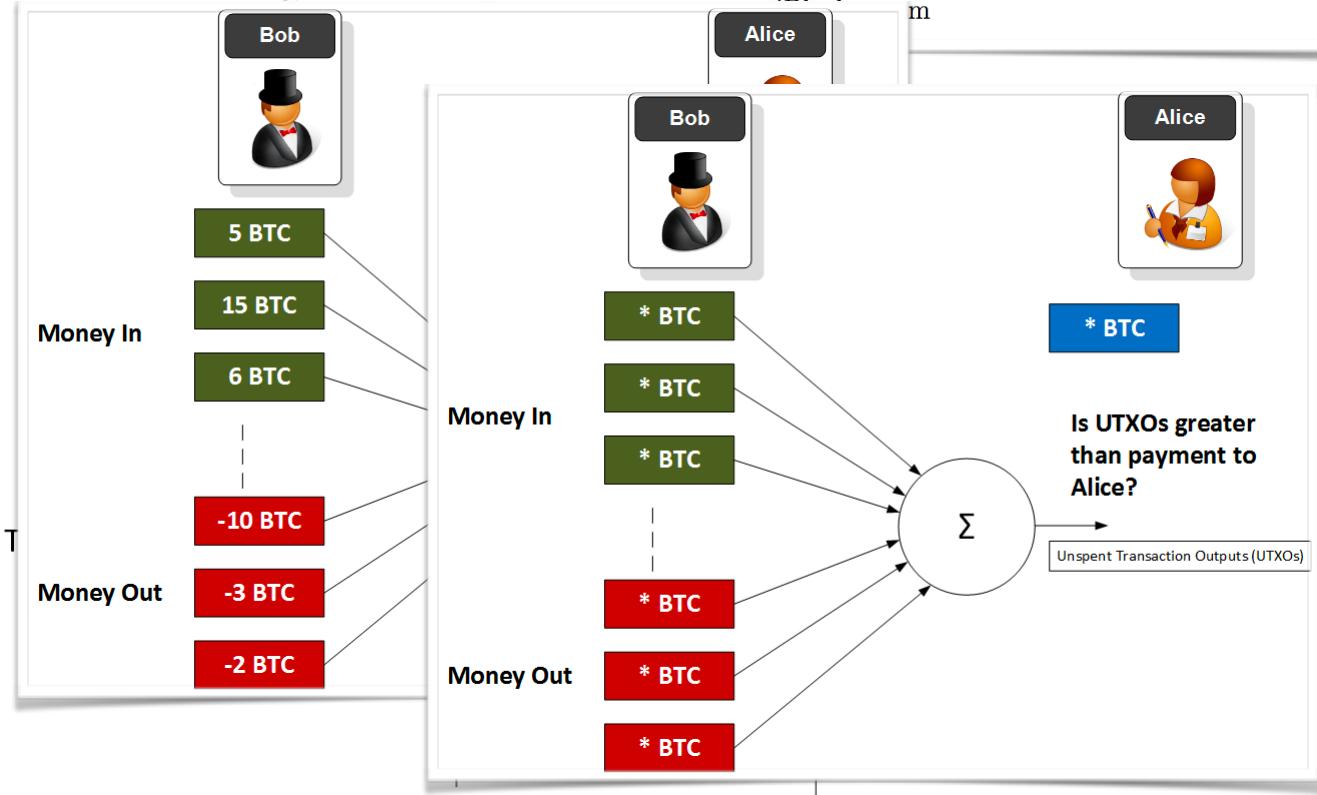
Bulletproofs: Short Proofs for Confidential Transactions and More

Benedikt Bünz^{*1}, Jonathan Bootle^{†2}, Dan Boneh^{‡1},
Andrew Poelstra^{§3}, Pieter Wuille^{¶3}, and Greg Maxwell^{||}

¹Stanford University

²University College London

m



Range Proof

Monero has seen an 80% reduction in transaction size, and which has also led to a significant reduction in the transaction fees.

Bob is applying for a loan from Alice, and now needs to prove that he has a salary of between \$60,000 and \$100,000, and that his employer — Trent — has at least \$1million in the bank ... merged into a single (and short) bullet proof



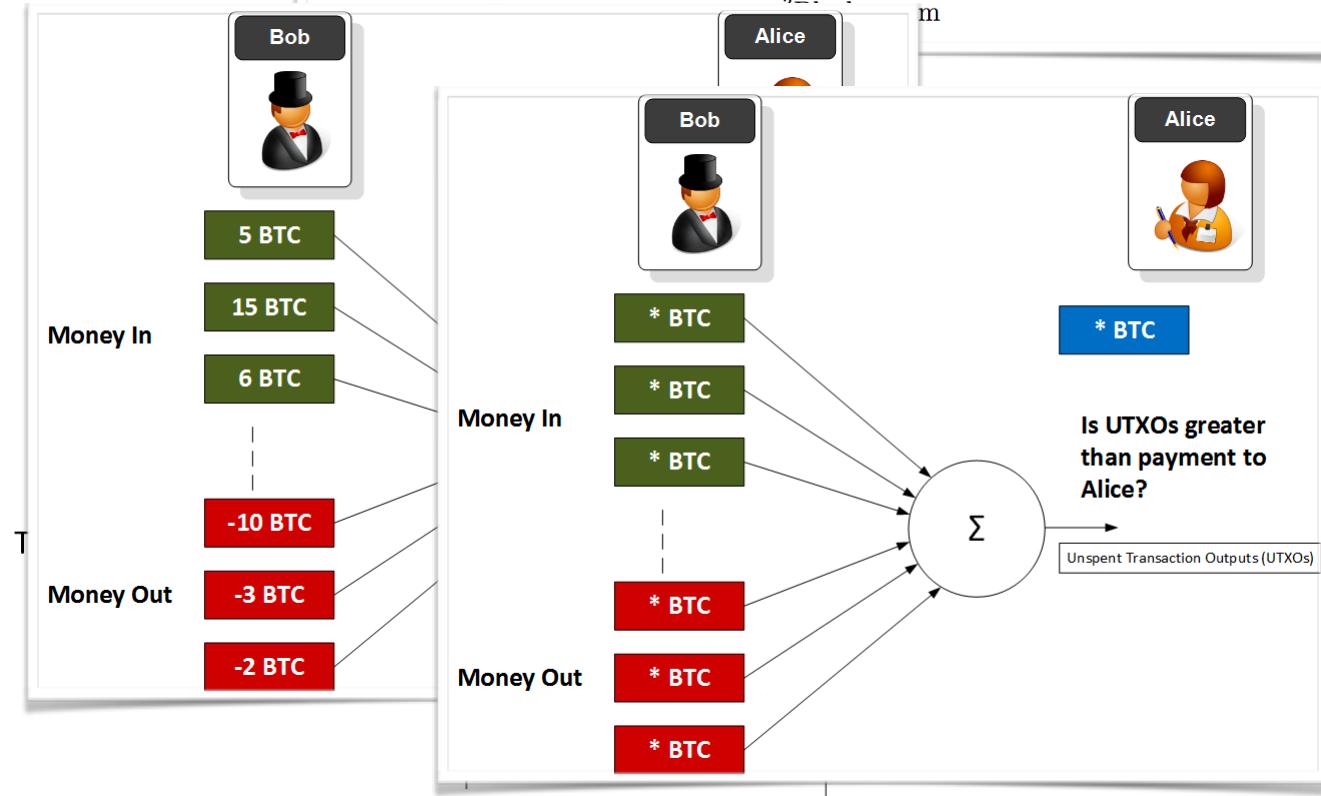
Bulletproofs: Short Proofs for Confidential Transactions and More

Benedikt Bünz^{*1}, Jonathan Bootle^{†2}, Dan Boneh^{‡1},
Andrew Poelstra^{§3}, Pieter Wuille^{¶3}, and Greg Maxwell^{||}

¹Stanford University

²University College London

³Monero



Range Proof

Monero has seen an 80% reduction in transaction size, and which has also led to a significant reduction in the transaction fees.

Bob is applying for a job from Alice, and wants to prove that his annual income is between \$60,000 and \$100,000, and that he has at least \$1million in savings... merged into a short bullet point.



Bulletproofs: Short Proofs for Confidential Transactions and More

Benedikt Bünz^{*1}, Jonathan Bootle^{†2}, Dan Boneh^{‡1}, Andrew Poelstra^{§3}, Pieter Wuille^{¶3}, and Greg Maxwell^{||}

¹Stanford University

²University College London

- Significant reduction in the size of the signature as opposed to other CT methods (such as zk-SNARKs and zk-STARKs).
- Significantly reduced transaction fees with shorter signatures.
- Supports MPC (Multiparty Computation) and where many parties can come together to create a single range proof, without revealing their secrets.
- Allow for the aggregation of range proofs and produces a single, and short, signature.
- Fast verification of proofs (and which are faster than most range proof methods, but still slower than zk-SNARKs).
- Design to be setup for blockchain integration.
- No need to setup a trust infrastructure. This often involves creating an initial set of encryption keys which are then used for trusted signatures. These keys should be used only once, and then deleted. If these keys are

Range Proofs

and More

- With bulletproofs, we can ask our financial institutions to prove that they have liquidity ... “Prove that you have more than \$1 billion of liquidity”, and if they failed to prove this, we would quickly move to audit them. Fraud on a large-scale basis would thus be detected in

University College London

- Significant reduction in the size of the signature as opposed to other CT methods (such as zk-SNARKs and zk-STARKs).
- Significantly reduced transaction fees with shorter signatures.
- Supports MPC (Multiparty Computation) and where many parties can come together to create a single range proof, without revealing their secrets.
- Allow for the aggregation of range proofs and produces a single, and short, signature.
- Fast verification of proofs (and which are faster than most range proof methods, but still slower than zk-SNARKs).
- Design to be setup for blockchain integration.
- No need to setup a trust infrastructure. This often involves creating an initial set of encryption keys which are then used for trusted signatures. These keys should be used only once, and then deleted. If these keys are

Monero has seen a significant reduction in the size of the signature, which leads to a significant reduction in the transaction fees.

Bob is applying for a job from Alice, and he wants to prove that his annual income is of between \$60,000 and \$100,000, and his employer — Trust — will receive at least \$1million. ... merged into a short bullet point.



Next Generation Crypto

Light-weight crypto.

Quantum-robust crypto

Tokenization.

Zero-knowledge.

Homomorphic Encryption.

zkSnarks, Range-proofs



Prof Bill Buchanan OBE

<http://asecuritysite.com/encryption>