Przegląd technologi blockchainu i bezpieczeństwa

Patryk Synowski, Dawid Pietrzak, Jakub Jastrzębski

Artykuł, z którym pracowaliśmy:

Contents lists available at ScienceDirect

Blockchain: Research and Applications

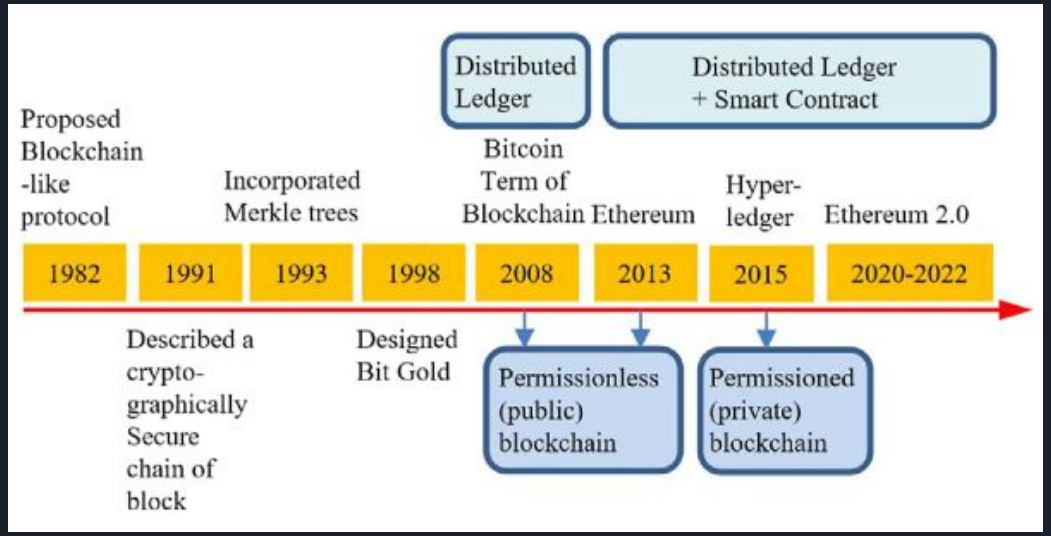
journal homepage: www.journals.elsevier.com/blockchain-research-and-applications

A survey on blockchain technology and its security

Huaqun Guo [a,*], Xingjie Yu [b,1]

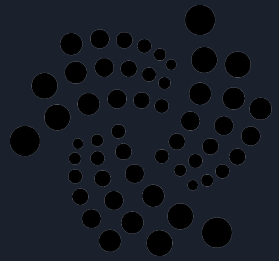
[a] Institute for Infocomm Research, A*STAR, Singapore
[b] Singapore

Szybka historia blockchainu



Algorytmy konsensusu

- Proof of Work (PoW)
- Proof of Stake (PoS)
- Delegated proof of stake (DPoS)
- Proof of elapsed time (PoET)
- Practical Byzantine Fault Tolerance
- Skierowany graf Acykliczny (DAG)



IOTA

Table 1
Comparison of consensus algorithms [13,37–40].

	PoW	PoS	DPoS	PoET	PBFT	DAG
Setup	Public permissionless/ Private blockchain	Public permissionless/ Private blockchain	Public/Private blockchain	Private permissioned/ permissionless blockchain	Private permissioned blockchain	Public permissioned non-blockchain
Cost of entry and returns	Relatively high cost of entry, but high returns	Low cost of entry, but low returns	Lower cost and lower returns than PoS	Very low cost of entry, but low returns	All participate with no return	All participate with no return
Incentives	The winning miner receives new coins with the block & transaction fees in the block he/she validates	The winner receives transaction fees with the new block. If a block winner attempts to add an invalid block, he/she loses his/her stake	The threat of loss of reputation & income provides an incentive for delegates to act honestly and keep the network secure	The winning miner receives the transaction fees with the new block he/ she validates.	Nil	Nil

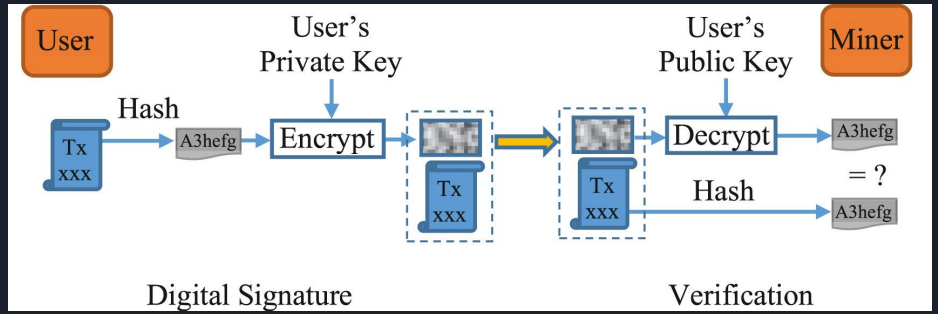
Cryptography of Blockchain

Aby użytkownicy mogli zaufać transakcjom przetwarzanym w blockchainie wykorzystywane są różne narzędzia kryptograficzne. Dzięki nim blockchain jest w stanie zapewnić zaufanie bez powierzania go jednej scentralizowanej instytucji.

Wykorzystywanymi mechanizmami kryptograficznymi są:

- Kryptografia klucza publicznego – podpisywanie transakcji
- Dowody z wiedzą zerową (Zero-knowledge proof) – udowodnienie posiadania informacji bez jej ujawniania
- Funkcje skrótu – łączenie ze sobą bloków transakcji, także element podpisu cyfrowego

Kryptografia klucza publicznego



- Wykorzystywana do weryfikacji autora transakcji
- Autor oblicza hash swojej transakcji, a następnie szyfruje go swoim kluczem prywatnym. Te zaszyfrowane dane są nazywane podpisem.
- Górnik odszyfrowuje podpis, uzyskując hash który porównuje z samodzielnie obliczonym hashem. Jeżeli te dane się zgadzają górnik ma pewność, że autorem transakcji jest właściciel klucza prywatnego i może dołączyć ją do następnego bloku.

Zero-knowledge proofs

Dowody z wiedzą zerową są wykorzystywane w kryptowalutach celem zapewnienia większej prywatności. Pozwalają one na udowodnienie komuś prawdziwości danego twierdzenia, bez udostępniania informacji na których oparte jest to twierdzenie. W przypadku kryptowalut pozwala to na weryfikację transakcji bez upubliczniania uczestników i przesyłanej kwoty.

- Deweloperzy Monero zastąpili wcześniejszy sposób obfuskacji transakcji (RingCT) mechanizmem “bulletproofs” aby zmniejszyć rozmiar transakcji.
- Zcash wykorzystuje “zk-SNARK” aby ukryć wartość oraz strony biorące udział w transakcji.

We can apply these ideas to a more realistic cryptography application. Peggy wants to prove to Victor that she knows the discrete log of a given value in a given group.[11]

For example, given a value y, a large prime p and a generator g, she wants to prove that she knows a value x such that $g^x \bmod p = y$, without revealing x. Indeed, knowledge of x could be used as a proof of identity, in that Peggy could have such knowledge because she chose a random value x that she didn't reveal to anyone, computed $y = g^x \bmod p$ and distributed the value of y to all potential verifiers, such that at a later time, proving knowledge of x is equivalent to proving identity as Peggy.

The protocol proceeds as follows: in each round, Peggy generates a random number r, computes $C = g^r \bmod p$ and discloses this to Victor. After receiving C, Victor randomly issues one of the following two requests: he either requests that Peggy discloses the value of r, or the value of $(x + r) \bmod (p - 1)$.

Victor can verify either answer; if he requested r, he can then compute $g^r \bmod p$ and verify that it matches C. If he requested $(x + r) \bmod (p - 1)$, he can verify that C is consistent with this, by computing $g^{(x+r) \bmod (p-1)} \bmod p$ and verifying that it matches $(C \cdot y) \bmod p$. If Peggy indeed knows the value of x, she can respond to either one of Victor's possible challenges.

If Peggy knew or could guess which challenge Victor is going to issue, then she could easily cheat and convince Victor that she knows x when she does not: if she knows that Victor is going to request r, then she proceeds normally: she picks r, computes $C = g^r \bmod p$ and discloses C to Victor; she will be able to respond to Victor's challenge. On the other hand, if she knows that Victor will request $(x + r) \bmod (p - 1)$, then she picks a random value r', computes $C' = g^{r'} \cdot (g^x)^{-1} \bmod p$, and discloses C' to Victor as the value of C that he is expecting. When Victor challenges her to reveal $(x + r) \bmod (p - 1)$, she reveals r', for which Victor will verify consistency, since he will in turn compute $g^{r'} \bmod p$, which matches $C' \cdot y$, since Peggy multiplied by the modular multiplicative inverse of y.

However, if in either one of the above scenarios Victor issues a challenge other than the one she was expecting and for which she manufactured the result, then she will be unable to respond to the challenge under the assumption of infeasibility of solving the discrete log for this group. If she picked r and disclosed $C = g^r \bmod p$, then she will be unable to produce a valid $(x + r) \bmod (p - 1)$ that would pass Victor's verification, given that she does not know x. And if she picked a value r' that poses as $(x + r) \bmod (p - 1)$, then she would have to respond with the discrete log of the value that she disclosed – but Peggy does not know this discrete log, since the value C she disclosed was obtained through arithmetic with known values, and not by computing a power with a known exponent.

Thus, a cheating prover has a 0.5 probability of successfully cheating in one round. By executing a large enough number of rounds, the probability of a cheating prover succeeding can be made arbitrarily low.

To show that the above interactive proof gives zero knowledge other than the fact that Peggy knows the x value, one can use similar arguments as used in the above proof of completeness and soundness. Specifically, a simulator, say Simon, who does not know the x value, can simulate the exchange between Peggy and Victor by the following procedure. Firstly, Simon randomly flips a fair coin. If the result is "head", he picks a random value r, computes $C = g^r \bmod p$, and discloses C as if it is a message from Peggy to Victor. Then Simon also outputs a message "request the value of r" as if it is sent from Victor to Peggy, and immediately outputs the value of r as if it is sent from Peggy to Victor. A single round is complete. On the other hand, if the coin flipping result is "tail", Simon picks a random number r', computes $C' = g^{r'} \cdot (y)^{-1} \bmod p$, and discloses C' as if it is a message from Peggy to Victor.

Przykład

- Posiadasz dwie kule, zieloną i czerwoną
- Twój kolega Wiktor nie rozróżnia koloru czerwonego od zielonego zatem te kule wyglądają dla niego identycznie
- Chcesz mu udowodnić, że kule są różne bez ujawniania która ma który kolor

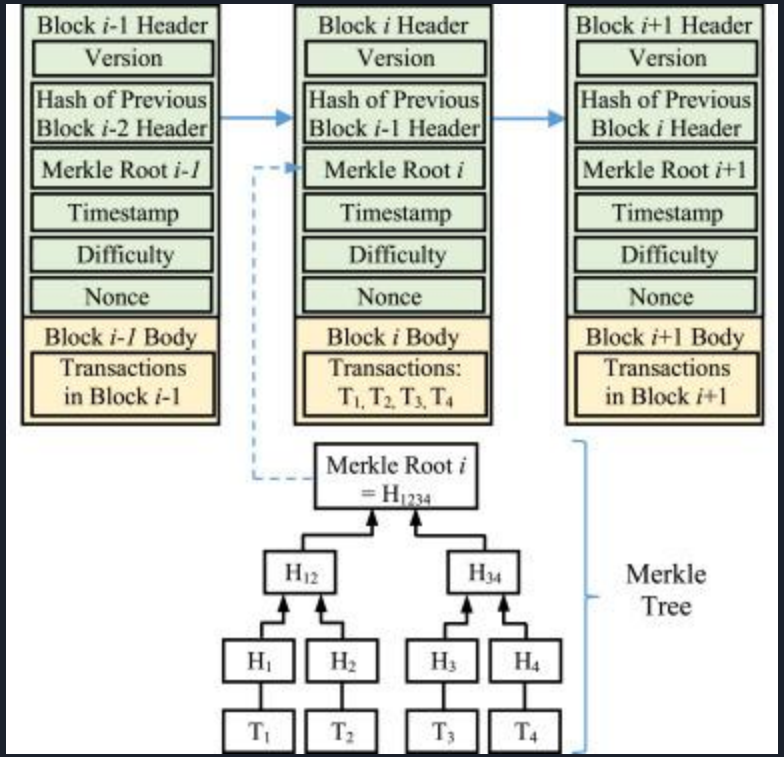
Dowód:

- Wiktor pokazuje ci jedną z kul, a następnie chowa je za swoimi plecami
- Następnie ponownie pokazuje jedną z kul, losowo decydując czy pokazać tą samą, czy tą drugą i zadaje pytanie “Czy pokazałem tą samą kulę?”
- Odpowiadasz tak lub nie. Wiktor jest w stanie zweryfikować tą odpowiedź
- Gdybyś nie był w stanie rozróżnić kul miałbyś 50% szans na poprawną odpowiedź
- Proces powtarzasz wielokrotnie, za każdym razem zwiększając prawdopodobieństwo że rzeczywiście jesteś w stanie rozróżnić te kule

Funkcje skrótu

Funkcje skrótu są wykorzystywane w

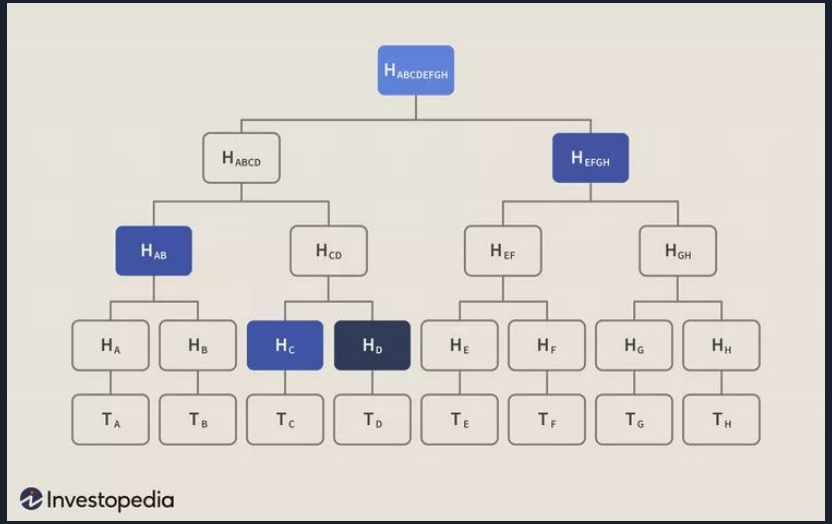
- podpisach cyfrowych
- do zapewnienia bezpieczeństwa historii transakcji w blockchainie
- do tworzenia adresów z kluczy publicznych
- do przyśpieszenia odnajdywania i weryfikacji historycznych transakcji


Block i-1 Header
Version
Hash of Previous Block i-2 Header
Merkle Root i-1
Timestamp
Difficulty
Nonce
Block i-1 Body
Transactions in Block i-1
Block i Header
Version
Hash of Previous Block i-1 Header
Merkle Root i
Timestamp
Difficulty
Nonce
Block i Body
Transactions: T1, T2, T3, T4
Block i+1 Header
Version
Hash of Previous Block i Header
Merkle Root i+1
Timestamp
Difficulty
Nonce
Block i+1 Body
Transactions in Block i+1
Merkle Root i = H1234
H12
H34
H1
H2
H3
H4
T1
T2
T3
T4
Merkle Tree

Drzewo Merkle

Drzewo Merkle przyśpiesza proces weryfikacji historycznych transakcji

- Użytkownik chce sprawdzić transakcję T_D, więc pobiera jej treść od innego uczestnika sieci
- Nie ma on jednak żadnego zaufania do jej treści
- Zamiast pobierać wszystkie inne transakcje w danym bloku, drzewo Merkle pozwala zweryfikować treść przy pomocy mniejszej ilości hashy (N vs Log N)



Zastosowania:

	Launched Year	Launched Price	Unit Price on Jan 1, 2021 (USD)	Unit Price on Feb 27, 2021 (USD)	Market capitalization on Feb 27, 2021 (USD)	Mined Numbers	Total Number
Bitcoin (BTC)	2009	0.0008 USD	28,994.01	47,781.33	890.6 billion	near 90%	21,000,000
Ethereum (ETH)	2014	Presale: 0.30 USD Homestead launched: 12.50 USD	737.71	1502	172.5 billion	114.84 M	Currently no implemented hard cap, & limited to 18 million per year
Cardano (ADA)	2017	0.019 EUR	0.31	1.36	43.4 billion	around 71%	45,000,000,000
Polkadot (DOT)	2020	1.2 USD	9.12	33.64	30.7 billion	1,049,328,830	Does not have a maximum supply
Litecoin (LTC)	2011	4.3 USD	124.67	176.31	11.8 billion	around 79%	84,000,000
Bitcoin Cash (BCH)	2017	543 USD	below 400	501.3	9.4 billion	near 89%	21,000,000
EOS	2017	2.29 USD	2.5975	3.68	3.5 billion	near 93%	1,027,393,754
IOTA	2016	Unknown	0.2969	1.1532	3.2 billion	2,779,530,283	2,779,530,283

Table 4
Blockchain security risk categories at low level in Ref. [29].

S/N	Category
1	51% vulnerability
2	Criminal activity
3	Private key security
4	Transaction privacy leakage
5	Double-spending
6	Criminal smart contracts
7	Under-priced operations
8	Smart contract's vulnerabilities
9	Under-optimized smart contract

Top 10 Web Application Security Risks	Assess on blockchain Technology	Analysis Examples
Injection	Poor input sanitization in blockchain technology	Before the EOS mainnet launches, discovered vulnerability of buffer-out-of-bounds write in EOS smart contract and the potential to run the malicious smart contract
Broken Authentication	A large attack surface exists without proper implementation of authentication functionality	The cryptocurrency LISK is an example of allowing an attack on authentication
Sensitive Data Exposure	High potential for this vulnerability	Vulnerable to data mining efforts—mining the public data on blockchain for useful information; Quantum computing will break the public key cryptography used to encrypt data on the blockchain
XML External Entities (XXE)	Not applicable	
Broken Access Control	One major vulnerability for smart contracts	Two attacks on Parity multi-signature wallets due to access control vulnerabilities
Security Misconfiguration	Affect blockchain security	Attackers exploited a vulnerability to steal cryptocurrency when Ethereum wallets were configured to receive external commands from port 8545
Cross-Site Scripting (XSS)	Affect blockchain in some ways	Blockchain explorers under XSS attack could display untrusted transaction data; Both blockchain explorers and wallets under XSS attack could allow access to a private key of a user and control over his/her account
Insecure Deserialization	May compromise of blockchain systems	If malicious users control transaction data, blockchain systems may be compromised by the vulnerable deserialization code
Using Components with Known Vulnerabilities	Very common to reuse code for Ethereum smart contracts	More than 90% of smart contracts in Ethereum did reuse code, and may contain known vulnerabilities
Insufficient Logging & Monitoring	The log owners may un-monitor their logs	May smart contracts lack of monitoring and hackers may exploit their vulnerabilities without being detected

Prawdziwe ataki i błędy w systemach blockchain

Core software bug

Attacks related to cryptocurrency exchange platforms

Attacks with wallets

Attacks and bugs with smart contract

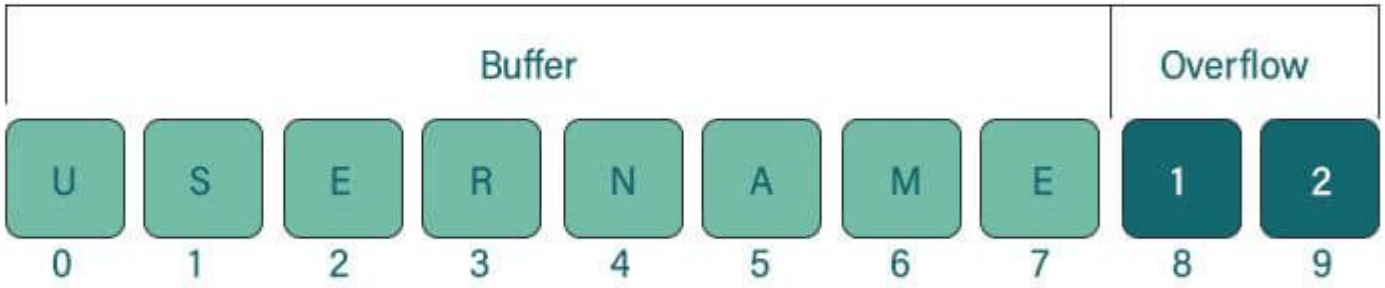
Network attacks

Endpoint attacks

Attacks with IOTA

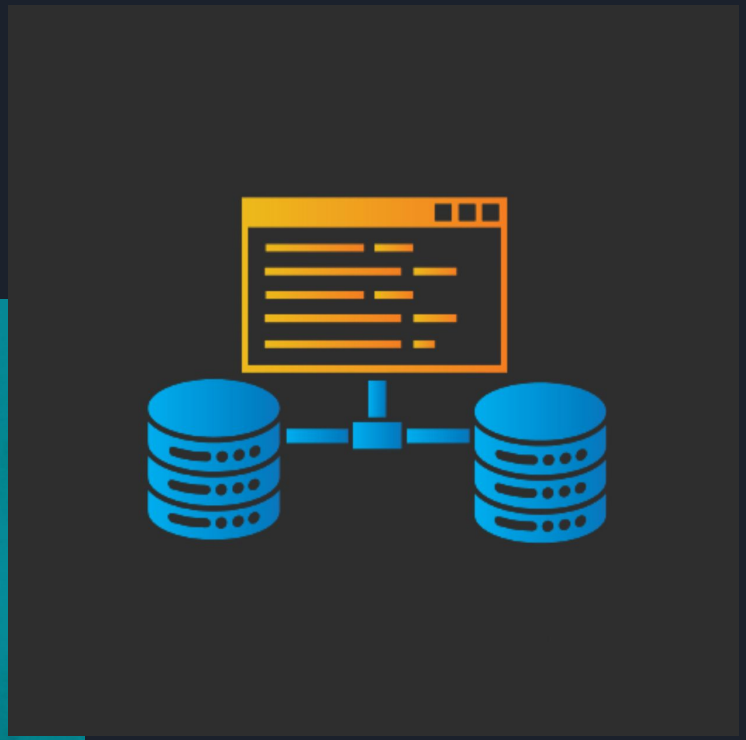
Core software bug

CVE-2010-5139
- Data opublikowania: Sierpień 2010
- Podatność miała miejsce w sieci Bitcoin, która z względu na lukę w protokole polegającą na przepełnieniu bufora pamięci (“buffer overflow”)

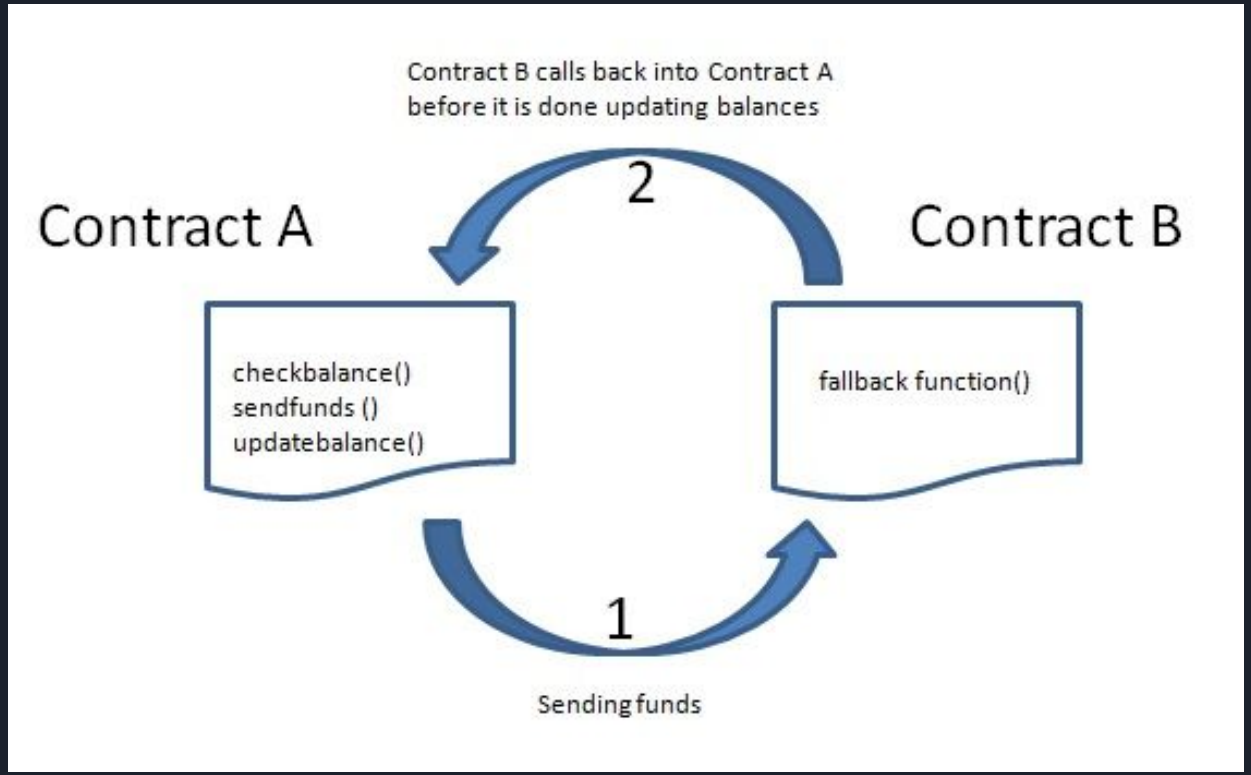


Attacks related to cryptocurrency exchange platforms

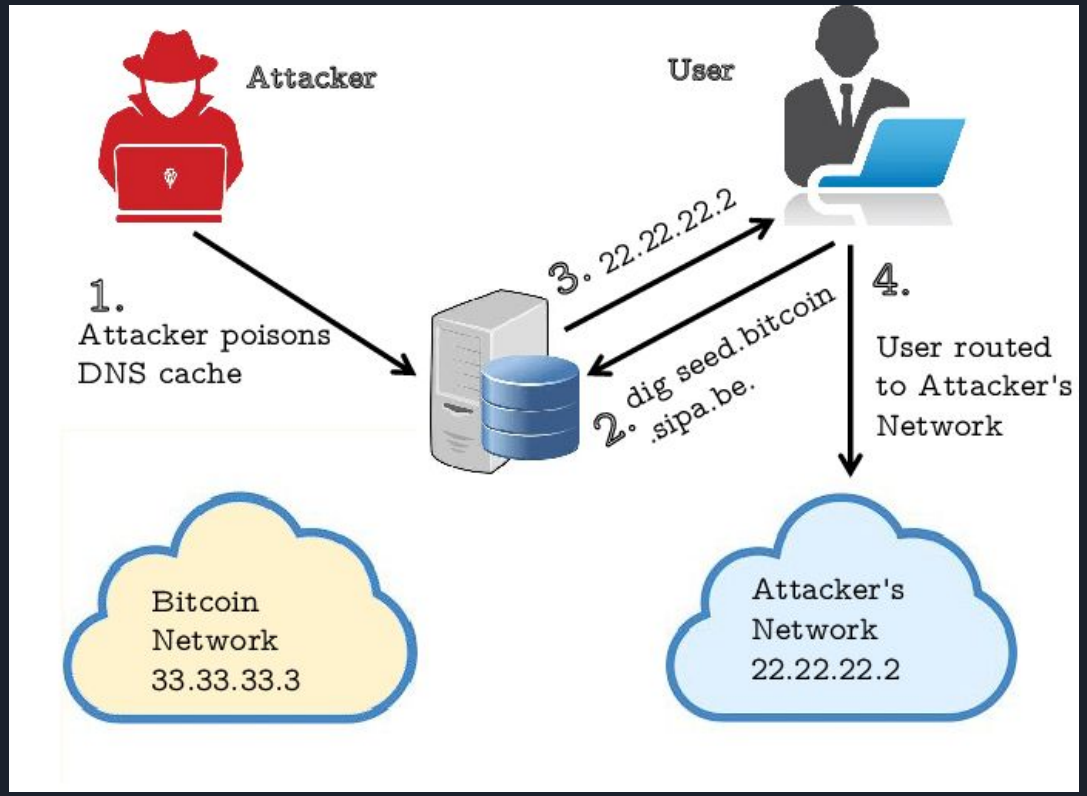
Attacks with wallets



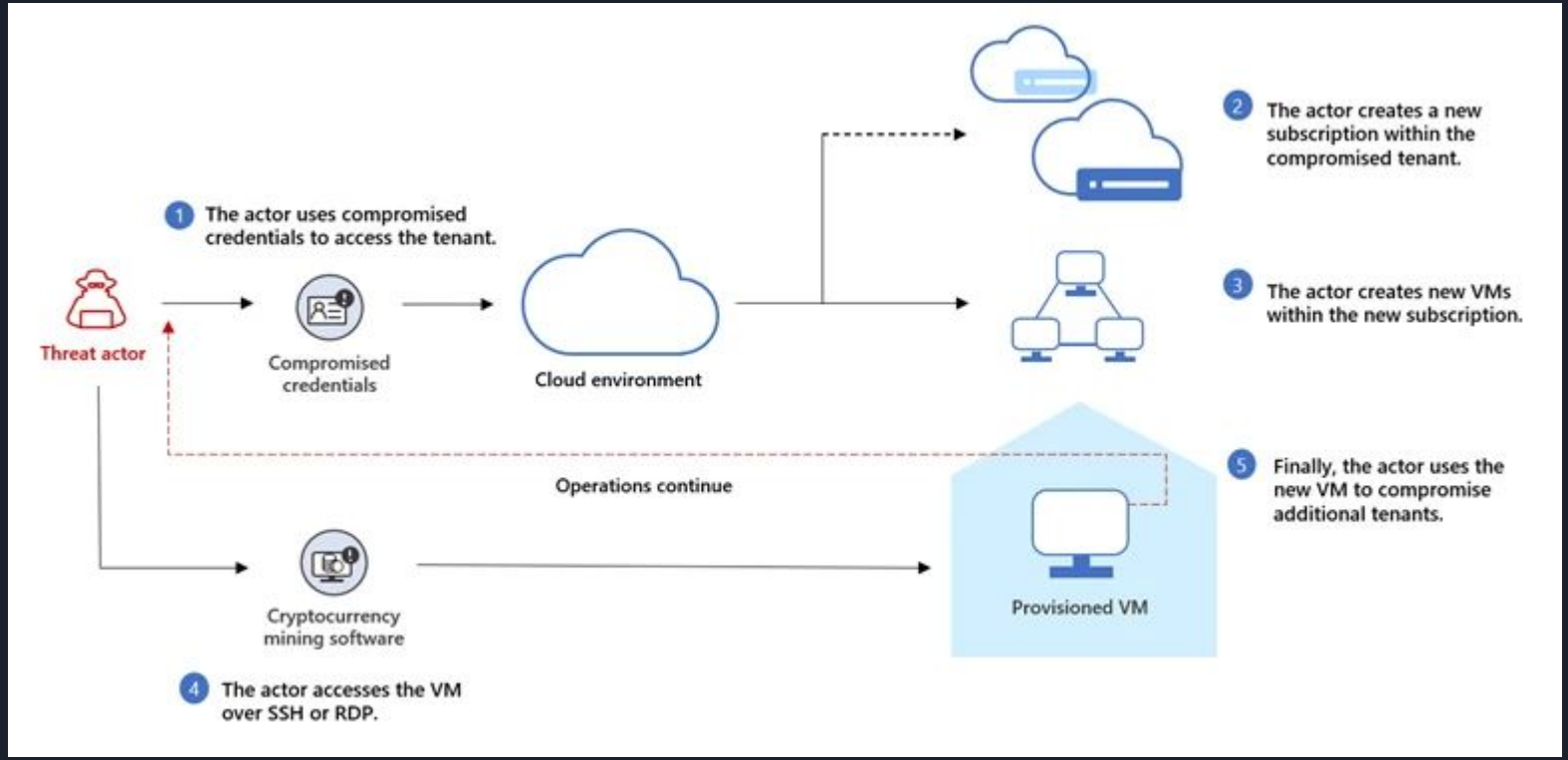
Attacks and bugs with smart contract



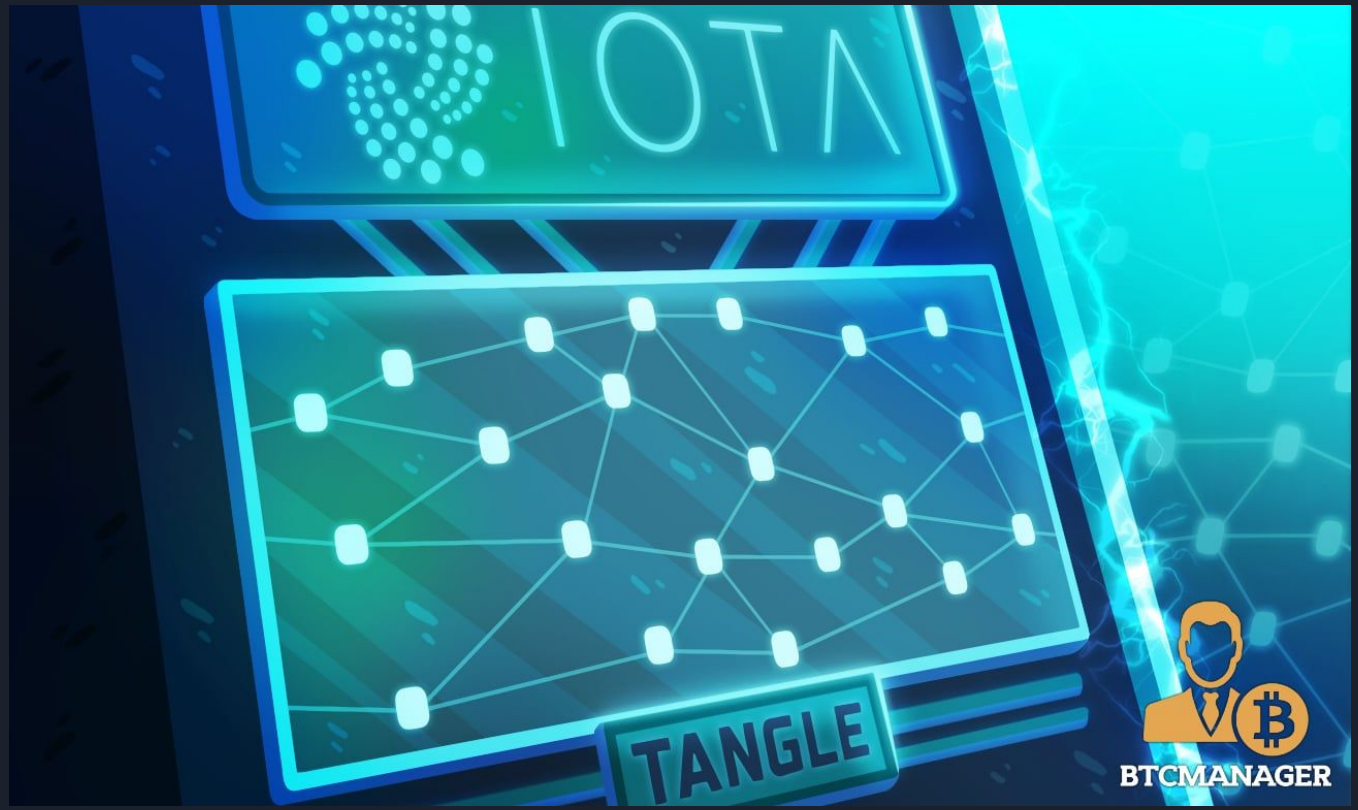
Network attacks



Endpoint attacks



Attacks with IOTA



Wyzwania i trendy

- Skalowalność? - BTC - 27 TPS, ETH ~ 1000s (PoS), EOS - 3996, PBFT - 3500, PoET- 2300
- Bezpieczeństwo smart kontraktów
- Regulacje i standaryzacja
- Komputery kwantowe
- Bezpieczeństwo IOTA