Blockchain: A Cautionary Viewpoint

Yvo Desmedt

Univ. of Texas at Dallas
USA

University College London
UK

July 11, 2018

Overview

1. The origins of cryptography
2. From Caesar to Alberti
3. From Alberti to Kasiski
4. From Kasiski towards foundations
5. The one-time pad
6. WWII: the birth of new superpowers
7. Shannon
8. Unconditional Security
9. The Vietnam war
10. Message Authentication
11. Hashed passwords
12. Public Key Cryptography

13. Consensus: the Byzantine problem
14. RSA signatures and hashing
15. Towards foundations for conditional security
16. Elliptic Curve Cryptography
17. Digital Timestamping
18. 1997: a dark year for authenticity
19. The breaking of several hash functions
20. Bitcoin and Blockchain
21. Problems with Blockchain
22. Alternatives to blockchains
23. Modern Research in cryptography
24. Conclusions

1. **The origins of cryptography**

In antiquity military leaders of empires needed to send letters back to the capital and were worried about spies.

So, the original goal of cryptography was the protection of secrecy. It was primarily used in diplomacy and military communication.

The “research” goes back to at least the Caesar cipher. (Note that other civilizations can make similar or older claims).

2. From Caesar to Alberti

Caesar cipher:

Caesar replaced each symbol in the original text, now called plaintext or cleartext, by one that was three positions further in the alphabet, counted cyclically. The word "plaintext", for example, would become "sodlqwhaw" in this system. The result is called ciphertext. The problem with this scheme is that anyone who knows how the text is encoded can break it. To prevent this, a key is used.

To describe a more modern variant of the Caesar cipher, let n be the cardinality of the alphabet being used, which is 26 for the English alphabet, or 27 when the space symbol is included in the alphabet. (In many old cryptoschemes, the space symbol was dropped since it would facilitate breaking the code.) The first symbol of the plaintext is mapped into the number 0, the second into 1, etc. To encrypt with the Caesar cipher, one adds modulo n the key k to the symbol m, represented as an integer between 0 and $n-1$. (Two integers a and b are equivalent modulo n, denoted as $a \equiv b \bmod n$, when a and b have the same non-negative remainder when divided by n). The corresponding symbol, then, in the ciphertext is $c = m + k \bmod n$, where the equality indicates that $0 \leq c < n$. If a long enough message contains redundancy, as plain English does, then an exhaustive search of all possible keys will reveal the correct plaintext. Decryptions (the process that permits the person who knows the secret key to compute the plaintext from the ciphertext) with the wrong key will (likely) not produce an understandable text. Since it is feasible to test all possible keys, the keyspace in the Caesar cipher is too small.

Substitution Cipher and Ciphertext-Only Attack:

We will now consider the substitution cipher. In the plaintext, each symbol m is replaced by the symbol $E_k(m)$, specified by the key k. To allow unique decryption the function E_k must be one-to-one. Moreover, if the same symbols are used in the ciphertext as in the plaintext, it must be a bijection. If the key can specify any such bijection, the cipher is called a simple substitution cipher. Obviously, for the English alphabet there are $26! = 403291461126605635584000000$, roughly $4 * 10^{26}$, different keys. We will now discuss the security of the scheme, assuming that the cryptanalyst only knows the ciphertext and the fact that a substitution cipher was used. Such an attack is called a ciphertext-only attack. Note that an exhaustive key search would take too long on a modern computer. Indeed, a modern parallel computer can perform 10^{17} operations per second. For simplicity, assume that such a computer could perform 10^{17} symbol decryptions per second. One wonders then how long the ciphertext needs to be before one can be certain that the cryptanalyst has found a sufficiently correct key. This measure is called the unicity distance. Shannon's theory of secrecy tells us that this is 28 symbols for an English text. An exhaustive key search would roughly take $3.6 * 10^3$ years before finding a sufficiently correct key. However, a much faster method for breaking a substitution cipher exists, which we will now describe.

In English the letter "e" is the most frequently used. Furthermore, no other letter has a frequency of occurrence that comes close to that of "e." A cryptanalyst starts the procedure by counting how many times each letter appears in the ciphertext. When the ciphertext is long enough, the most frequent letter in the ciphertext corresponds to the letter "e" in the plaintext. The frequencies of the letters

“T,O,A,N,I,R,S,H” are too similar to decide by which letter they have been substituted. Therefore the cryptanalyst will use the frequency distribution of two or three consecutive letters, called a digram and a trigram. When the space symbols have been discounted, the most frequent digrams are: “th”; “e” as the first letter, decreasing in order as follows: “er,ed,es,en,ea”; and “e” as the second letter: “he,re.” The digram “he” is also quite common. This permits the identification of the letter “h”, and then the letter “t”. The next step is to distinguish the vowels from the consonants. With the exception of the diagrams “ea,io” two vowels rarely follow one another. This allows one to identify the letter “n”, since 4 out of 5 letters following “n” are vowels. Using similar properties of other digrams and trigrams, the full key is found. If mistakes are made, they are easily spotted and one can recover using backtracking.

A note:

Since encryption (i.e. transforming the plaintext into ciphertext) was done by hand or later mechanically, the schemes that could be used were heavily restricted.

3. From Alberti to Kasiski

We start by quoting Kahn's 1966 book (see p. 41) on polyalphabetic ciphers:

> Leo Battista Alberti devised a new cryptographic principle that lies at the basis of most modern ciphers. Called polyalphabeticity, it employs a number of cipher alphabets for a single message. Alberti generated his several alphabets from a single primary one by sliding them in relation to the conventional alphabet.

A special, and well known case, is the so called Vigenère cipher (first invented by Giovan Battista Bellaso) in which all aforementioned ciphers are the same and they all are the Caesar cipher. We briefly explain this scheme more mathematically. We assume the plaintext and ciphertext are over an alphabet with n characters. We view the plaintext as a tuple $(m_0, m_1, \ldots, m_{l-1})$, where l is the length of the plaintext. A key now is a short tuple $(k_0, k_1, \ldots, k_{s-1})$, which will remain fixed during the duration of the use of the cipher. Typically $s < l$. To encrypt, we map the alphabet into Z_n. The Caesar cipher then correspond to: mapping a plaintext character and the key to Z_n, which result we call m'_i and k', the corresponding ciphertext $c'_i := m'_i + k' \bmod n$. The result is then mapped back to the alphabet. In the case Vigenère cipher is used, k is replaced by $k_{i \bmod s}$ and so the ciphertext "character" $c'_i := m'_i + k'_{i \bmod s}$.

We now survey the cryptanalysis of such polyalphabetic ciphers and its impact on historic ciphers. We again quote Kahn's 1966 book (see pp. 41-42):

> For some 300 essentially years this cipher system remained impregnable.

4. From Kasiski towards foundations

Again quoting Kahn:

> Friedrich Kasiski, a retired German' infantry major, discovered and in 1863 published the general method for the solution of polyalphabetic ciphers with repeating keys.
>
> . . .
>
> Kasiski's technique for breaking a polyalphabetic cipher stimulated cryptographers to devise more ingenious enciphering schemes. They proposed using keys that did not repeat ("running keys"), such as the text of a book.

Kahn surveys the cryptanalysis of the running key cipher in his book (p. 42) as following:

> In 1883, however, a French language teacher named Auguste Kerckhoffs devised the general solution for polyalphabetic ciphers. His

technique is called superimposition.

The cryptanalyst hunts among his intercepted messages for two or more identical ciphertext fragments. These would indicate that the same portion of running key has enciphered a repeated fragment of a plaintext. Another way to discover where the same portion of running key has been used to encipher two or more messages is by solving the system of indicators by which one cipher clerk tells another where in the keybook he is starting the running key.

The cryptanalyst then writes out the messages, lining them up so that parts that have been enciphered by the same portion of the running key stand one below the other . . . . This will assemble into columns the letters that have been enciphered by the same key letter; the result is columns of letters that have each been enCiphered monoalphabetically. The cryptanalyst subjects each column to frequency analysis and thereby recovers the plaintext.

5. The one-time pad

The one-time pad and Shannon's analysis of its security is one of the most important discoveries in modern cryptography. We will first discuss the scheme, then give a formal definition of the security, and discuss its security.

The Scheme

In Vernam's one-time pad:

- the key is (at least) as long as the message. Before encrypting the message the sender and receiver have obtained a secret key, a string, of which the symbols have been chosen uniformly random in the set S and independent.

- the message is a string of symbols belonging to the alphabet (a finite set) S, for example $\{0,1\}$, on which a binary operation "$*$" is defined, for example the exor (exclusive-or). We assume that $S(*)$ forms a group.

Let m_i, k_i and c_i be the i^{th} symbols of, respectively, the message, the key and the ciphertext, each belonging to S.

The encryption algorithm produces $c_i := m_i * k_i$ in S.
To decrypt, the receiver computes $m_i = c_i * k_i^{-1}$.

Mauborgne suggesed to use the key only once. This implies that if a new message needs to be encrypted a new key (i.e., chosen independently from the previous) is chosen, which explains the terminology: one-time pad.

It is trivial to verify that this is an encryption scheme. In the case $S(*) = Z_2(+) = \{0, 1\}(+)$, the integers modulo 2, the encryption algorithm and the decryption algorithm are identical and the operation corresponds with an exor (exclusive or).

Note that due to inexpensive storage, the one-time pad is now in many circumstances practical.

6. WWII: the birth of new superpowers

Before WWII the USA and USSR were insignificant. In the 1904-1905 war the Russian fleet was virtually annihilated. When Roosevelt became president, the US Army was smaller than the one of Sweden.

The breaking of the Japanese Naval Codes allowed the USA to set a trap in Midway and eventually win WWII. Similarly in Europe, the breaking of Engima and the Lorenz cipher shortened the war for the Western Allies dramatically.

7. Shannon

Definition 1. Shannon defined an encryption system to be perfect when, for a cryptanalyst not knowing the secret key, the message m is independent of the ciphertext c, formally:

$$\text{prob}(\mathbf{m} = m \mid \mathbf{c} = E_k(m)) = \text{prob}(\mathbf{m} = m). \tag{1}$$

Theorem 1. *(Shannon) The one-time pad is perfect.*

WWII encouraged this research. It inspired the first A/D and D/A converters. It was used immediately between Churchill and Roosevelt. Shannon's paper was kept classified until after WWII. The implementation was classified for even longer.

8. Unconditional Security

Since the one-time pad scheme cannot be broken (except when Einstein is right), the security of these schemes have been called unconditional (other names: information theoretic).

(Note that some people: call these "non-cryptographic". Since Shannon was the first scientific researcher in the field of cryptography, anyone calling such schemes non-cryptographic is undermining Shannon contribution to cryptography and so the use of "non-cryptographic" should be strongly discouraged).

9. The Vietnam war

The US code used during the Vietnam war was allegedly broken by the Hungarians. (Source: personal communication Dénes, 1994).

10. Message Authentication

The second scientific paper on cryptography was published by Gilbert-McWilliams-Sloane in 1974. It was inspired by the Cuban crisis and the fact US Generals recommended President Kennedy to have nuclear war with the Soviet Union because “only” 50 million US Citizens would die.

The scheme uses projective geometry, but it is trivial to make an affine version of this scheme. The security is unconditional.

Kennedy wanted to avoid that a nuclear strike could be started by a disgruntled US General. The problem was given to Sandia Labs, and Simmons asked researchers around the world in combinatorics whether they could find a solution.

(Around 1980 work was improved dramatically by Wegman-Carter, using universal Hash Functions, and later in Scandinavia by e.g., Johansson and Smeets).

Note that Universal Hash Functions are different from the cryptographic hash functions used in Blockchains. The Universal ones are unconditionally secure and use a key.

11. Hashed passwords

The idea of hashing is to have a many-to-one mapping.

Non-cryptographic hashing was used in sorting and searching, well before cryptographic hashing was used in information security.

Purdy in 1974 proposed the use of hashing for password storage.

Before Purdy passwords were stored in the clear (there are still systems that do this, or similarly) and then the system manager could read the passwords. The system manager could then login using somebody's account, making the auditing of inside attacks more difficult.

When the password is hashed, the system manager does not learn the password. He can change the password, but that will leave a trace!

Note: Purdy's proposed hash is not secure. Moreover, when the entropy of the passwords is low, then hashing does not help that much.

12. Public Key Cryptography

Many regard the invention of public key cryptography (Merkel, 1978) and Diffie-Hellman (1976) as having solved the key-distribution problem. However, the limitations of their work were made clear by researchers from Computer Security, who pointed out that:

- The public key might not be authentic (Popek and Kline, 1979). (The solution requires a PKI, and today's implemented solution is far from perfect.)
- To deny a digital signature, one can claim the secret key has been stolen (Saltzer, 1978).

Since digital signatures are not used for the purpose these were invented, the last criticism is academic.

13. Consensus: the Byzantine problem

Lamport-Shostak-Pease (1982) introduced the first paper on consensus.

A battle can only be won when all armies either attack or retreat.

The problem becomes worse since some generals may be dishonest, by telling some they want to retreat while telling others they want to attack.

It was shown that when the parties that can be dishonest are bounded by a threshold t and the protocol is synchronous that at least $3t + 1$ parties are needed. In the protocol all honest generals will come to the same conclusion.

14. RSA signatures and hashing

RSA signatures were published in 1978. We had to wait until 1982 before Davida published his multiplicative attack. RSA signatures as described in the original 1978 paper suffered from the following problem:

$$\text{Signature}(M_1 \cdot M_2) = \text{Signature}(M_1) \cdot \text{Signature}(M_2)$$

Several variants of this attack were published that had to be addressed before RSA signatures could be used in real life applications.

Denning in 1984 proposed the use of hashing to deal with this attack. Today cryptographic hashing is the standard method used before using a public key system to sign.

Note: padding is often also needed and some padding scheme were vulnerable to a multiplicative attack (see Coron-Naccache-Stern 1999).

15. Towards foundations for conditional security

A proper foundation for conditional security may have to wait until serious progress has been made on computational complexity, which may take centuries.

In the meanwhile, Goldwasser-Micali (see also Rabin, etc.) proposed the concept of proven security, which work as follows:

- One makes a security definition to express (using mathematics and computational complexity) against what attack one wants to protect.
- One develops a scheme, one hopes will satisfy the required security.
- One attempts to prove the proposed scheme satisfies the security definition, using an unproven computational complexity.

- One tweaks the draft scheme/protocol to obtain a scheme/protocol which is proven secure. Often this tweak reduces the practicality of the scheme.

Note that:

- lately, some researchers tweak the security assumption to be able to prove their schemes to be secure. This gives rise to questionable assumptions that are not properly analyzed.
- the development of “practical” proven secure schemes/protocols often took decades.

Zero-knowledge and simulatability: Goldwasser-Micali-(Rackoff) developed these concepts in the context of interactive proof. Today many security proofs use the concepts of zero-knowledge and simulatability.

(Note that the concept of zero-knowledge interactive proofs immediately found several applications in smart card technology and digital signatures.)

16. Elliptic Curve Cryptography

Koblitz and independently Miller (1985) proposed the use of elliptic curves for the use of cryptography.

From an algebraic viewpoint, this is nothing else than proposing to use a different group.

ElGamal signatures (1985) and the Schnorr's (1989) more practical variant can trivially be used over elliptic curves.

All schemes used before 1991 were broken by Menezes-Vanstone-Okamoto (1991). The schemes used today are different than these. There has been no major progress on breaking the elliptic curve systems in use today, except when using a quantum computer, which some regard as hypothetical.

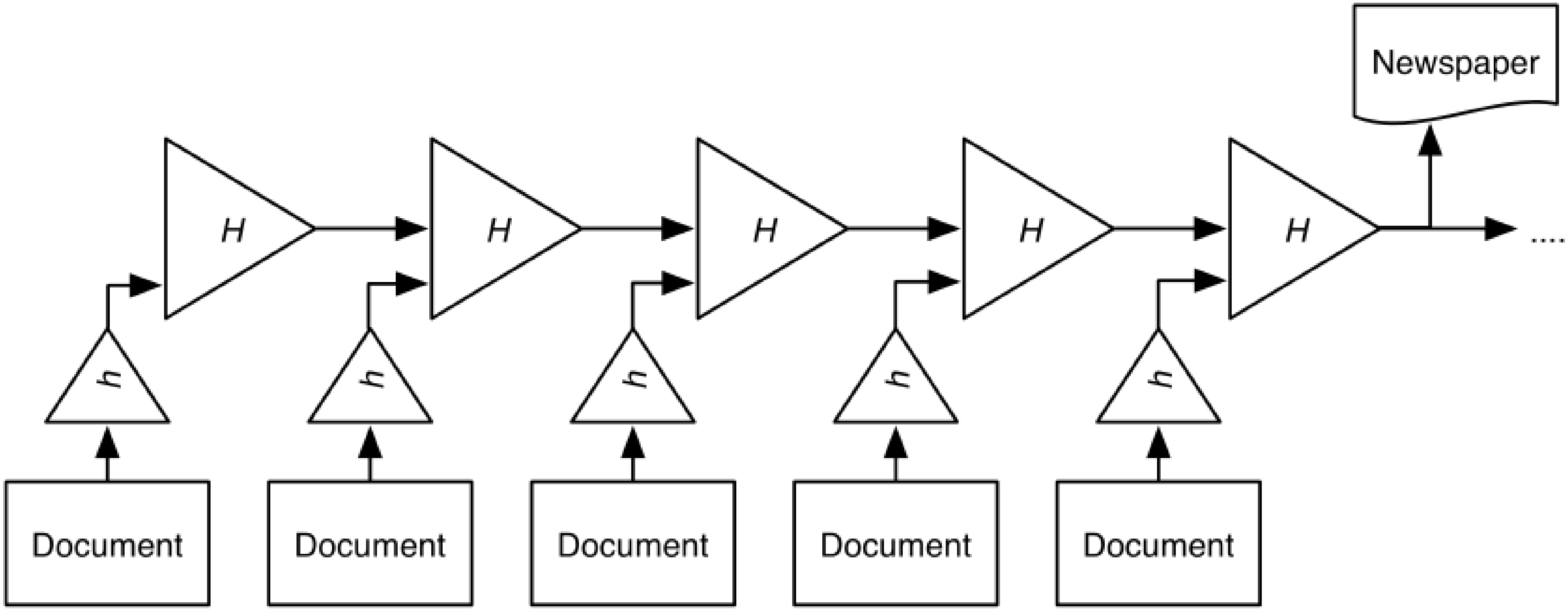
Elliptic curve systems got a boost to their use:

- in old-fashioned mobile (cell)-phones and then later smartphones,
- Vanstone
- NSA and NIST

17. Digital Timestamping

Haber-Stornetta (1991) suggested the use of digital timestamping.

They suggested (among other ideas) to use Linear Hash Chain:



©Wikipedia

18. 1997: a dark year for authenticity

Since 1997 US Governmental research on authentication is classified:

Impact:

- Soon afterwards several researchers left Sandia Lab,
- One can wonder whether the US Government knew that SHA-1 was insecure,

19. The breaking of several hash functions

In 2004 Wang and her co-authors shocked the cryptographic community when the heavily used cryptographic hash functions MD4 (Rivest, 1990) and MD5 (Rivest, 1992) got broken.

For MD5 a collision (i.e., finding two values hashing in the same output) was found in 2004, that used differentials computed by hand and then using an IBM p690 cluster. Some organizations ignored the attack and so we had to wait until March 2005, when Lenstra-Wang-de Weger demonstrated construction of X.509 certificates of public keys that hashed in the same result.

In February 2005 Wang-Yin-Yu started undermining SHA-1 by presenting a theoretical attack against the full version of SHA-1. Due to this attack SHA-3 was developed. An actual collision against SHA-1 was demonstrated by Stevens in February 2017.

SHA-2 was developed by NSA in 2001. SHA-2 and older hash functions use the MerkleDamgård construction, which has problems with quantum computers (see higher).

SHA-3 was selected from several competing proposals. NIST announced the winner in 2012 (standard since August 201s). SHA-3 does not use the MerkleDamgård attack and no efficient quantum algorithmic attack is known.

20. Bitcoin and Blockchain

Blockchain was invented by “Satoshi Nakamoto” in 2008 in the context of the cryptocurrency bitcoin. Chaum’s digital cash needed a bank for each transaction. That was to “prevent” double spending. Two approaches were considered by Chaum (and later co-authors):

- the first that ran to the bank got the money.
- the identity of a double spender is revealed.

The use of blockchain avoided this double spending problem.

Main idea: use hashing as proof of work. (Note: proof of work is older.)

Details: solve a “puzzle” by finding an x_i such that $H(x_{i-1}, x_i) = 0$.

Validating the block can be done using digital signatures.

The problem of forks:

Forks can occur when:

- two different valid x_i are found!
- one did not agree on a state (see later)
- one x_i is invalid.

Solution: the longest chain wins.

21. Problems with Blockchain

We just said: in case of a fork, the longest chain wins.

Implications:

- the one with the most energy wins! (For the use of bitcoin, this approach already consumes more electricity than Switzerland).

- there is a long delay before one agrees that the chain is long enough. This delay makes blockchain problematic in many applications

Other problems with blockchain:

- anything can be written in a blockchain. (The case of child pornography made the news in March 2018).

- Privacy: several problems such as un-erasability and Big Brother becoming Huge Brother.

- Cryptographers often ignore computer insecurity and many computers use the same platform. So, a weakness in a platform might be exploited on many computers! So, a majority rule for giving permission may not be optimal.

22. Alternatives to blockchains

Many alternatives have been proposed. We only focus on:

- Proof of stake: the next block is chosen based on a possible combinations of different methods.

  Avoiding centralization is important.

- Permission: Consensus

23. Modern Research in cryptography

Seeing the importance of blockchains and the use of hash functions, we would believe that a large part of the community is trying to break or improve the primitives.

However we see that:

- the research on cryptanalysis has decreased in importance. Indeed:
  - While at Crypto in the 1980's 33-48% of the papers were on cryptanalysis, at Crypto 2012 it was only 6%.
    (Recently Alexander May stated that this has now increased to 15%.)
  - NSF (USA) is not funding any major research on cryptanalysis.

- The topics under research do not always match with the current need.

  The clearest examples of this are anonymity and privacy! To give two examples:

  - Despite 35 years research on secure multiparty computation

    No Key Application of Secure multiparty computation has been found

    (Smart, 2017.)

  - Despite Europe's GDPR, many users decided to allow Facebook extensive use of data.

  Conclusion: for most users privacy is not so important. Research on "extreme" privacy is receiving millions of dollars in funding.

- Cryptographic multilinear maps: an obsession

Halevi in his CRYPTO 2015 invited talk:

The State of Cryptographic Multilinear Maps

suggested that the research community should copy the practical cryptographic community by dropping security proofs!

This may unfortunately have set back the research in cryptography by many decades and has lowered the standard of papers accepted at major venues such as Eurocrypt, indeed we have:

1 The paper: Shai Halevi. ”Graded encoding, variations on a scheme”. Cryptology ePrint Archive, Report 2015/866, 2015. Available at https://eprint.iacr.org/2015/866.

The attack: Jean-Sbastien Coron, Moon Sung Lee, Tancrde Lepoint and Mehdi Tibouchi. "Cryptanalysis of GGH15 Multilinear Maps". In CRYPTO 2016.

2 The paper: Garg, Sanjam and Gentry, Craig and Halevi, Shai, "Candidate Multilinear Maps from Ideal Lattices", EUROCRYPT 2013, pp. 1–17

The attacks:

a. Jung Hee Cheon, Jinhyuck Jeong, and Changmin Lee. "An algorithm for CSPR problems and cryptanalysis of the GGH multilinear map without an encoding of zero". Technical report, Cryptology ePrint Archive, Report 2016/139, 2016.

b. Yupu Hu and Huiwen Jia. "Cryptanalysis of GGH map". In Advances in Cryptology - EUROCRYPT, pp. 537-565, 2016.

c. Eric Miles Amit Sahai Mark Zhandry. ”Annihilation Attacks for Multilinear Maps: Cryptanalysis of Indistinguishability Obfuscation over GGH13”. In CRYPTO, 2016.

d. Daniel Apon and Nico Döttling and Sanjam Garg and Pratyay Mukherjee. ”Cryptanalysis of Indistinguishability Obfuscations of Circuits over GGH13”. In 44th International Colloquium on Automata, Languages, and Programming (ICALP 2017).

3 The paper Craig Gentry, Sergey Gorbunov and Shai Halevi. ”Graph-Induced Multilinear Maps from Lattices”. TCC 2015. The attack: Jean-Sbastien Coron, Moon Sung Lee, Tancrde Lepoint and Mehdi Tibouchi. ”Cryptanalysis of GGH15 Multilinear Maps”. In CRYPTO 2016.

Conclusion: cryptographic multilinear maps have become the El Dorado of cryptography. Expeditions to find it were a huge waste of money.

Note that:

– the search for fast public key system was given up after roughly 2 years.

– open problems, are good to make progress in a field. However, accepting dubious papers is not the way to go. No serious journal published an incorrect proof of Fermat's Last Theorem.

24. Conclusions

We see that:

- Making consensus practical is an important problem.
- No cryptosystems has lasted more than 300 years.
- It may take centuries before we solve fundamental questions in computational complexity theory and fundamental questions in conditionally secure cryptography.
- In the last 15 years we have seen that several hash functions, which are the foundation of blockchain technology, have been totally broken.
- In contrast to other primitives, such as public key cryptography and

block ciphers, hash functions seem to be the weakest.

- Why would the USA, that became a superpower by breaking cryptosystems, tolerate standards it can not break?

- Can one combine existing hash functions to get stronger primitives?

  Note: an attempt was made for block ciphers, but the claim is incorrect.