CrypTorino 2021

TITOLO SERIE

Book Series

XX

CrypTOrino 2021

Proceedings of the CrypTO Conference 2021 and related topics

a cura di

Laura Capuano

Guglielmo Morgari

Lea Terracini

Contributi di

Aracne editrice

www.aracneeditrice.it
info@aracneeditrice.it

www.gioacchinoonoratieditore.it
info@gioacchinoonoratieditore.it

via Vittorio Veneto, 20
00020 Canterano (RM)
(06) 45551463

isbn 000-00-000-0000-0

edizione: xx 9999

Indice

Parte I
Extended Abstracts

Parte II
Invited Surveys

CrypTOrino 2021
ISBN 000-00-000-0000-0
DOI 10.4399/00000000000001
pag. 9–?? (xx 9999)

Introduction

Nowadays cryptography plays a central role in the security of our digital world. Although normally not perceived by users, cryptography is in fact at the heart of a number of operations we routinely perform every day: withdrawals from ATM, mobile phone calls, home banking and online purchases, to name just a few examples, strongly rely on cryptographic techniques to guarantee user security rights like confidentiality and privacy. Despite its highly applicative nature, cryptography has solid theoretical foundations in different areas of mathematics, including abstract algebra, number theory, geometry, probability, complexity theory and information theory. A deep understanding of these theoretical aspects and their link with application problems is thus fundamental to use cryptography in a correct and effective way.

This volume collects some proceedings of the first CrypTo Conference, organized in May 2021 by the Cryptography and Number Theory Group of Politecnico di Torino and Università di Torino with the aim of giving an overview of the current directions in cryptography. In addition to the proceedings of the conference, two surveys are also included in the volume, authored by researchers strictly connected by scientific collaborations to the above-mentioned group. All the presented works have undergone a blind review process in order to guarantee high quality and conformance to the scope. The authors represent both academia and industry and come from numerous countries (Italy, United States, England, United Arab Emirates), thus providing different and heterogeneous views on current research trends in cryptography.

The topics addressed in the volume are many and closely interrelated, covering both theoretical and practical aspects. They concern innovative technologies of great practical interest such as blockchain and distributed ledgers; more classic but still fundamental subjects such as the integer factorization problem and related cryptosystems; the Post-Quantum world faced from different points of view; abstract subjects like cryptographic algebraic tools and finally fundamental cryptographic primitives.

More in detail, blockchains and distributed ledgers (DLT) represent a highly topical and concrete subject, which is likely to radically and irreversibly change the operating model of many traditional and innovative businesses related to data processing. Although this technology is known above all for cryptocurrencies, it actually has countless possible applications. Examples of financial and data storage applications are presented in the volume. A somewhat controversial aspect of blockchain technology, that is the ability to (pseudo)anonymize users, is also considered with an analysis of the most promising deanonymization techniques based on Machine Learning.

Threshold Signatures, discussed in the volume, represent a very important research topic, due to their theoretical depth and possible applications. Among these, Threshold Signatures role in cryptocurrency custody services is highlighted, testifying to the close relationship between theory and application.

Although Quantum Computer is now publicly available only at a prototype level, it is generally believed that in the next few years it might be able to break the public key systems currently in use. Consequently, a topic of great importance in the world of research is the definition of quantum-resistant solutions. As is well known, two deeply distinct but complementary solutions are today considered: Post Quantum Cryptography (PQC) and Quantum Key Distribution (QKD). A survey on the ongoing process by NIST dedicated to PQC, aimed to define cryptographic standards for the next decades, is reported in the volume. Furthermore, a scheme of ring signatures is presented, based on isogenies between elliptic curves, mathematical functions arising from algebraic geometry whose computations are currently believed to be inviolable by Quantum Computer. QKD, normally studied in its aspects of quantum mechanics, is instead dealt with in the volume from the complementary and equally important point of view of information theory.

The volume also presents some more theoretical contents, enlightening the close relationship of cryptography with other branches of mathematics. Among these are considered algebraic aspects such as the study of polynomial maps over finite fields, at the base of many cryptographic techniques and relevant in many other mathematical areas as well. In addition, classical topics such as the Integer Factorization Problem and the RSA cryptosystem are reconsidered and innovative contributions are provided in terms of characterization and generalization towards more efficient and safe solutions. Finally, the volume contains a survey on the One Time Pad (OTP), the only cryptographic solution that is unconditionally secure and which, for reasons of efficiency, is often approximated by more practical but necessarily imperfect solutions.

Parte I

EXTENDED ABSTRACTS

CrypTOrino 2021
ISBN 000-00-000-0000-0
DOI 10.4399/00000000000002
pag. 15–?? (xx 9999)

DeFi 2020: the bank-less revolution

Andrea Di Nenno

Permissionless Decentralised Financial applications have been dominating the blockchain space in 2020, from the Ethereum ecosystem later spanning across many other blockchains. Conceived, designed and implemented during the crypto winter between 2018-2019 several protocols for Stable-coins, Lending, Borrowing, Exchanging crypto and tokens were deployed on main-net and started to get traction: Total Value Locked (TVL) in those protocol, after reaching the historical milestone of 1 billion in March 2020, has exploded to 50 billion USD within the first year [1]. The revolutionary nature of this technology is twofold.

From a use-case point of view, Decentralised Finance eliminates any barrier, especially in terms of law, regulations and intermediaries, for any individual around the world to access and personally harness from a global decentralised financial system, running 24/7, where laws and mechanics are written in open source code that any user can access before interacting and at the same time providing users more control over their money through personal cryptographic wallets. DeFi products have transparency by default; as not only are they built upon open-source technology, but every transaction and interaction between users and applications is recorded in an open, immutable ledger distributed around the world. While it might be months or years before a centralized cryptocurrency exchange is discovered to have gone insolvent, DeFi's solvency and health is always subject to the collective observation and analysis of a large open-source community where anyone can point out fraud and systemic risk. This clearly is opposed to CeFi (Centralised Finance), where services are typically opaque or subject to information asymmetry, with the public being provided with much less insight than what held by the infrastructure. This creates unknown levels of risk exposure at the same time entrusting the risk management to a small group of regulators. An example of this is from 2008: many mortgage-related financial products were considered incredibly safe by few big players like Moody's Investors Service, Standard and Poor's, and Fitch Ratings, until it was revealed they were insufficiently collateralized, triggering a global financial crisis.

On the tech side, given that decentralisation requires all the information to be publicly accessible in the ledger, the protocols running on it are effectively applications that store their internal state and expose APIs to the other users, being smart contract or Externally Owned Accounts. These protocols are then composable by nature, meaning that they possibly can inter-operate between each other without requiring adapters. This allows for truly impartial and deterministic applications that run as coded and are incapable of being shut down. Ethereum composability has lead to a sharp organic growth of the entire system, where basic dApps effectively enables more complex protocols and structures to be build on top of those, creating a positive feedback loop that reminds early day of Internet: as Internet grew in users, the incentive for building on it grew, while the obstacles shrank. The permission-less nature of public blockchains, where anyone can deploy its own protocol that can interact with all the existing ones, amplified the magnitude of competition which in turn lead to higher quality of the systems and in some cases strong and collaborative communities.

1. Stable Coins

One of the most important key enablers to replicate and innovate the financial system on blockchain was the creation of stable-coins, that is crypto assets that aren't subject to volatility like other cryptocurrencies but keep their value stable. On blockchain systems this can be easily achieved via centralisation, simply via an external entity holding fiat currencies and minting blockchain native stable coins, supposedly at 1:1 ratio. While this is the most convenient way to obtain a stable-coin, clearly introduces systemic risks with centralisation. Rather more difficult is to obtain a truly decentralised stablecoin, that is an asset that keeps its value stable through a decentralised protocol.

MakerDAO protocol [2] is so far one of the most succesful project, designed for years since 2014 and finally deployed on Ethereum mainnet by the end of 2017. Its stablecoin Dai, can be minted by anyone by simply providing collateral, in form of crypto, at a 150%

ratio. Dai is only generated by depositing collateral assets into the Maker Protocol smart contracts. Once generated, it can be used in the same manner as any other cryptocurrency: it can be sent to others, used as payments for goods and services, and even held as savings through more complex features. Every Dai in circulation has the requirement to be directly backed by excess collateral, meaning that the value of the collateral is higher than the value of the Dai debt, keeping its value pegged to 1$. A complex system made up of data oracles, keepers and insurances is required to maintain this healthy ratio between Dai in circulation and collateral value, which is governed through a DAO (Decentralised Autonomous Organisation), whose participants are solely Maker token MKR holders. At the time of writing, the TVL in MakerDAO is worth around 10 billion dollars with a total supply of 3.5 billion Dai [3].

2. Automated Market Makers

With the explosion of ERC-20 tokens on Ethereum during the ICO bull market in 2017, the need to have a medium to natively swap them through an on-chain protocol was clear. It was the dawn of Decentralised Exchanges, also known as Automated Market Makers, where it's not a central broker to make the market, matching bids and asks, but open protocols running on blockchain systems. The first greatest breakthrough in this space was given by Hayden Adams, at the time mechanical engineer, that created Uniswap [5]. The idea was to have a protocol that allows liquidity provider to deposit a pair of assets in pools that are then used by users to swap one token with the other at a price that follows a constant product curve of the quantity of tokens in the two pools, such that $x * y = k$. Traders pay 0.3% fees back to the pool which is split between the liquidity providers. Since 2017, Uniswap protocol has grown enormously in terms of users, liquidity and integration, up to a point where it generates billions of volume everyday and millions in fees for liquidity providers.

3. Lending and Borrowing

Automated Market Makers as a way to obtain decentralised token swaps enabled more complex financial instruments to be built leveraging this simpler primitives. Open lending and borrowing protocols arose, surely lead by Compound [6] and Aave [4]. These comes as decentralised non-custodial liquidity market protocol where users can participate as depositors or borrowers. Depositors provide liquidity to the market to earn a passive income, while borrowers are able to borrow from the available liquidity against their collateral. A game of incentives is built to maintain the systems solvent at all times, ensuring no user is in a position where it cannot repay his debt to the borrower.

A ground breaking idea was brought by Aave with so-called flash-loans: under-collateralized loans that a borrower can take with the requirement that they must be payed back within the same transaction. If not payed, the whole transaction, including the borrowing of the liquidity would be reverted to previous state, only making the borrower pay fees. This native financial primitive, made possible due to the atomicity of blockchain transactions, allows potentially everyone to borrow a big amount with zero collateral, exercise a trade, pay back the debt and keep the profit, with the only requirements being funds to pay for the whole transaction.

4. Derivatives Markets

As many knows, the biggest financial markets are the derivatives markets, which are estimated to float at hundreds of trillions of dollars. This certainly is constituting the trend in DeFi from summer of 2019, with Synthetix Protocol [7] that was the first to bring Synthetic Assets on-chain: instruments that only tracks the value of an external (underlying) asset of whatever kind. A complex protocol of debt pools, data oracles and liquidation systems allows user to gain exposure with their crypto to price movements of external assets such as gold, stocks or fiat currencies. During the last couple of years, many other derivatives protocol have been successfully running and attracting users and liquidity, spanning from insurances protocols [8], options [9], prediction markets [10], and perpetuals.

CrypTOrino 2021
ISBN 000-00-000-0000-0
DOI 10.4399/00000000000003
pag. 19–19 (xx 9999)

Bibliografia

[1] DeFi Pulse - The Decentralised Finance Leaderboard, https://defipulse.com/

[2] MakerDAO - An Unbiased Global Financial System, https://makerdao.com/

[3] Dai statistics, https://daistats.com/

[4] AAVE - Open source DeFi Protocol, https://aave.com/

[5] Uniswap - A fully decentralized protocol for automated liquidity provision on Ethereum, https://uniswap.org/

[6] Compound - autonomous interest rate protocol, https://compound.finance/

[7] Synthetix - derivatives liquidity protocol https://synthetix.io/

[8] Nexus Mutual - decentralised alternative to insurance, https://nexusmutual.io/

[9] Hegic - On-chain non custodial options, https://www.hegic.co/

[10] Augur - peer-to-peer, decentralized prediction market platform, https://augur.net/

Clearmatics
dinennoandrea@gmail.com

CrypTOrino 2021
ISBN 000-00-000-0000-0
DOI 10.4399/00000000000004
pag. 21–?? (xx 9999)

An overview of blockchains' de-anonymization attacks

Andrea Gangemi

Blockchains were first described in 2008, when Satoshi Nakamoto published a paper outlining the first cryptocurrency, Bitcoin [1]. Bitcoin attracted the attention of researchers and practitioners, thanks to its lack of a central authority and its supposed high level of pseudoanonymity. In fact, a person does not insert any personal data when interacting with the blockchain, but he or she receives a pseudonym, usually referred as an address, which is obtained hashing the public key of the user [2].
A user can send some of his cryptocurrency to another user, thanks to transactions: they are formed by a list of input addresses that refer to previous unspent transactions outputs (UTXO), and a list of output addresses. One of the output addresses is usually a change address and it belongs to the same user that issues the transaction.

The UTXO model is easy to understand and represent, however the privacy of the network decreases if it is not used correctly.
In fact, a smart use of Machine Learning (ML) algorithms can de-anonymize the blockchain address space. Usually, the first applied technique is clustering, which is typically based on the idea that all the input addresses in a transaction belong to the same user [3]. These clusters cannot be reconducted to a real identity, unless some specific details about them are obtained thanks to off-chain information. This heuristic is nowadays not so accurate, after the inception of mixing services like CoinJoin - however, it is still the best way to discriminate between different addresses.

ChainAnalysis [4] offers to interested researchers already labeled clusters, where each label corresponds to one prominent blockchain activity (e.g. exchange or gambling). Sun Yin et al [5], starting from labeled clusters, had the idea to use supervised learning techniques in order to predict the label of uncategorized clusters. They compared various ML algorithms that were ranked thanks to the F1-score metric. The best performing algorithms were Gradient Boosting Classifier and Random Forests.

A different and more modern approach represents the blockchain network like a weighted graph to try to predict if a specific address belongs to an exchange or not [6]. The network can be seen like a bipartite graph, where vertices depict both transactions and addresses. The key idea takes advantage of the concept of motif, a small subgraph with statistical significance. A N-motif is simply a path of length $2N$ in the bipartite graph. [7] tries to extend the work, predicting with a generalization of this approach up to five blockchain activities: exchange, gambling, mining, service and darknet. This is the first step towards graph representation learning, a novel ML field that uses the network structure of the underlying data to improve predictive outcomes. As of today, there are not a lot of works in this research area: one example is given by [8].

In response to these well-known anononimity problems, cryptocurrencies with privacy-enhancing techniques have been deployed. A first example is Monero [9], launched in 2014. Monero does not identify the real input being spent, because it is mixed together with other inputs, called mixins. Each transaction uses a ring signature that is valid just for the real input, but it does not reveal any information about which one it is.
However, at first a transaction was considered valid even without additional mixins, and for this reason most users deployed transactions with exactly one input. [10] designed some heuristics that were able to reveal in most situations the real input even for transactions that used at least one mixin as suggested by the privacy recommendations. Monero deployed a hard fork which forced every user to use the same number of mixins, together with their new transaction standard, RingCT. Monero kept updating the protocol roughly every six months: after 2018, there have been no successful attacks towards the network.

A second example is ZCash [11], launched in 2016 as a fork of the Bitcoin protocol. It supports two kinds of addresses: t-addresses, that behave similarly to bitcoin addresses, and z-addresses, which are private and do not reveal the coins (ZEC) that have been spent thanks to the use of zk-SNARKs, a zero-knowledge algorithm.
There are four possible types of transactions on ZCash. Since the

t-addresses can be treated as bitcoin addresses, they can easily be clustered and labeled. The interesting idea of this blockchain is of course the shielded pool: coins can be moved from a t-address to a z-address, or in the other way around. However, most users do not use z-addresses correctly, and this behaviour reduces the global privacy of the network. Indeed, if a t-to-z transaction is linked to a z-to-t transaction, we can reduce the size of the anonimity set, damaging the users which are trying to get more privacy. [12] showed some easy heuristics to link most of these transactions.

Different approaches must be used to de-anonymize account-based blockchains, like Ethereum [5]. This model works like a bank account: if we send a transaction, Ethereum subtracts its value from the balance of our account, and it adds the same value to the balance of the recipient. Every account is linked to an Ethereum address. We are therefore forced to issue multiple transactions using the same input address. Since every Ethereum transaction has exactly one input and one output, we cannot use the clustering techniques described to de-anonymize the UTXO-based cryptocurrencies. However, similar ideas can be utilized in some specific cases.
[14] tries to leverage an approach to explore why people deploy smart contracts on the blockchain. Since it is a good practice in informatics to reuse code, contracts can be compared and grouped together using specific distance measures, like Levenshtein distance. To label a group, the researchers used the four most frequent words in the contracts contained in that cluster. They discovered that the majority of smart contracts are used to generate tokens or can be reconducted to scams or Ponzi schemes.
[15] uses graph representation learning techniques to link Ethereum addresses owned by the same users. They investigated three metrics: the active time of the day, the gas price selection and the location in the Ethereum transaction graph. They tested twelve algorithms, and the best were Diff2Vec [16] and Role2Vec [17]. They used this approach to de-anonymize some trustless mixing services, and they showed how, using simple heuristics together with these ML algorithms, they could restrict the real size of the anonimity set.

With the development of new ML algorithms, more powerful attacks are expected in the next years. Users should start using privacy-enhancing wallets: for example, Bitcoin users might download Wasabi [18]. In this way, clustering techniques would be less effective and cryptocurrencies could really be considered pseudoanonymous.

CrypTOrino 2021
ISBN 000-00-000-0000-0
DOI 10.4399/00000000000005
pag. 25–26 (xx 9999)

Bibliografia

[1] Nakamoto, S., Bitcoin: A peer-to-peer electronic cash system, http://bitcoin.org WhitePaper (2008).

[2] Antonopoulos, A., Mastering Bitcoin: Unlocking Digital Cryptocurrencies, O'Reilly (2017).

[3] Ermilov, D. et al, Automatic Bitcoin Address Clustering, 16th IEEE International Conference on Machine Learning and Applications (ICMLA), 461-466 (2017).

[4] https://www.chainalysis.com/

[5] Sun Yin, H. et al, Breaking Bad: De-Anonymising Entity Types on the Bitcoin Blockchain Using Supervised Machine Learning, Hawaii International Conference on System Sciences 2018, 3497-3506 (2018).

[6] Ranshous, S. et al, Exchange Pattern Mining in the Bitcoin Transaction Directed Hypergraph, Financial Cryptography and Data Security, 248-263 (2017).

[7] Jourdan, M. et al, Characterizing Entities in the Bitcoin Blockchain, IEEE International Conference on Data Mining Workshops (ICDMW), 55-62 (2018).

[8] Zhang, Q. et al, Bitcoin Transaction Forecasting with Deep Network Representation Learning, IEEE Transactions on Emerging Topics in Computing, to Appear.

[9] van Saberhagen, N., CryptoNote v 2.0, https://www.getmonero.org/ WhitePaper (2013).

[10] Möser, M. et al, An Empirical Analysis of Traceability in the Monero Blockchain, Proceedings on Privacy Enhancing Technologies, 143-163 (2018).

[11] Sasson, E. et al, Zerocash: Decentralized Anonymous Payments from Bitcoin, IEEE Symposium on Security and Privacy, 459-474 (2014).

[12] Kappos, G. et al, An Empyrical Analysis of Anonymity in ZCash, 27th USENIX Security Symposium, 463-477 (2018).

[13] Buterin, V., Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform, https://ethereum.org/en/WhitePaper (2013).

[14] Norvill, L. et al, Automated Labeling of Unknown Contracts in Ethereum, 26th International Conference on Computer Communication and Networks (ICCCN), (2017).

[15] Béres, F. et al, Blockchain is Watching You: Profiling and Deanonymizing Ethereum Users, arXiv:2005.14051 (2020).

[16] Rozemberczki, B, Sarkar, R., Fast sequence based embedding with diffusion graphs, Complex Networks IX, 99-107 (2018).

[17] Ahmed, N. et al, Learning Role-based Graph Embeddings, arXiv:1802.02896 (2018).

[18] https://wasabiwallet.io/

Politecnico di Torino
andrea.gangemi@polito.it

CrypTOrino 2021
ISBN 000-00-000-0000-0
DOI 10.4399/00000000000006
pag. 27–?? (xx 9999)

Filecoin: from Proof-of-Space blockchain to decentralized storage.

Irene Giacomelli

1. Introduction

In the recent years, we have witness an increasing research interest in systems that can be used to decentralize the internet. The goal is designing and implementing distributed protocol that can replace centralized services and trusted parties. For instance, Bitcoin [4] and Ethereum [5] are examples of using a blockchain network to get decentralized transaction ledgers.

In the same line of work, we find Filecoin [6]: a decentralized storage network that turns cloud storage into an algorithmic market. The market runs on a blockchain with a native protocol token (also called "Filecoin"), which miners earn by providing storage to clients. Conversely, clients spend Filecoin hiring miners to store or distribute data. As with Bitcoin, Filecoin miners compete to mine blocks with sizable rewards, but Filecoin mining power is proportional to active storage, which directly provides a useful service to clients (unlike Bitcoin mining, whose usefulness is limited to maintaining blockchain network consensus). This creates a powerful incentive for miners to amass as much storage as they can, and rent it out to clients.

2. Proof of Space

One of the core building block of Filecoin is a new cryptographic primitive called Proof of Space (PoS) and introduced in 2015 by [1]. Informally, a PoS is a 2-party protocol where a party, called the verifier, uses a small amount of storage and computation to check that the other player, the prover, stores some data of size N. We require that the algorithm executed for the verification is highly efficient, whereas for the prover computing the proof is highly efficient providing he stores and has access to the data. More formally:

Definition 1. A Proof of Space (PoS) is an interactive protocol between two random access machines, the prover P and the verifier V, defined by two sub-protocols:

— Initialization: V and P with common input an identifier id and a storage bound $N \in \mathbb{N}$ run an interactive protocol that outputs a string S (called advice) of size N. From now on, S is stored by P. This phase is executed only once, as a setup. Note that V does not need to store S, but only a short commitment to it (eg, hash of S).

— Execution: V and P with input S run an interactive protocol at the end of which V outputs a bit, 1 means V is convinced that P stores S. This phase can be repeated many times to check P's storage.

These two protocols are designed in such a way we have the following properties:

— Completeness. If P and V follow the protocols, then V outputs 1 during an execution phases with overwhelming probability.

— Soundness. A PoS is (N_0, T)-sound if a verifier interacting with a malicious prover $\tilde{P}$ with the following 2 constrains outputs 1 during execution with negligible probability. Constraints:

a) $\tilde{P}$'s persistent storage after the initialization is $\leq N_0$ (ie, $\tilde{P}$ persistently stores an advice S that has size $\leq N_0$);
b) $\tilde{P}$'s running time during the execution is $\leq T$.

The value $\epsilon = (N - N_0)/N$ is called the spacegap.

PoS constructions with a negligible spacegap can be obtained using the graph-labeling paradigm. Given a direct acyclic graph and an hash function H, define the label of a node as the hash of the node index concatenate with the labels of the parents. Then, the initialization phase is defined as running the full labeling algorithm to get the labels of the sink nodes. This is the advice S stored by the prover. For such constructions, we can prove the soundness property by studying the underlying graph and showing that its topology guarantees a computation-space trade-off: given n pre-computed (eg, stored) labels, the labeling functions requires at least $T(n)$ steps (ie, calls to H). See for example the Stacked-DRGs graph in [2].

3. PoS in Filecoin

In the Filecoin system, to get mining power a miner posts on chain (a commitment to) the advice S. Then, to maintain its power, the miner has to repeatedly and publicly answer to new instances of the execution protocol accordingly to a precise schedule. The schedule is decided in such a way that we can argue that for a successfully miner the running time is $\leq T$ and therefore persistent storage of the advice is guaranteed.

In Filecoin we are interested in useful space, that is storage space that is used to keep real-world data. Therefore, we want that the advice S of the PoS to encode some real data D instead of just being a random incompressible sequence of bytes. A simple way to achieve this goal is the following: add the input D (data) for the prover in the initialization phase and make the advice S a function of $Comm_D$, a commitment to the data known also by the verifier. The new advice stored by the prover, called the replica, is defined as $R = S + D$. What we get with this construction is a proof of useful space that is actually a Proof of Replication [2]. This means that in addition to the property seen before, the replica R has the extraction property. This guarantees the existence of an extraction algorithm that can recover the original data D from the interaction with a successful prover during the execution phases. In [3], it is proved that these two properties together, soundness plus extraction, gives the rational replication property. That is, a miner in Filecoin asked to store two of more copies of the same data, does not save storage if it decide to makes the replicas not independent. In other words, storing the data in a replicated format is the rational strategy.

CrypTOrino 2021
ISBN 000-00-000-0000-0
DOI 10.4399/00000000000007
pag. 31–31 (xx 9999)

Bibliografia

[1] Dziembowski, S., Faust, S., Kolmogorov, V., Pietrzak, K. Proof of Space, Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference Proceedings, Part II. Lecture Notes in Computer Science 9216, Springer (2015).

[2] Fisch, B. Tight Proofs of Space and Replication, Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques Proceedings, Part II. Lecture Notes in Computer Science 11477, Springer (2019).

[3] Fisch, B. PoReps: Proofs of Space on Useful Data.
https://eprint.iacr.org/2018/678.

[4] Bitcoin, `https://bitcoin.org/en/`

[5] Ethereum, `https://ethereum.org/en/`

[6] Protocol Labs, Filecoin whitepaper, July 19 (2017).
`https://filecoin.io/filecoin.pdf`

Protocol Labs
irene@protocol.ai

CrypTOrino 2021
ISBN 000-00-000-0000-0
DOI 10.4399/00000000000008
pag. 33–?? (xx 9999)

Threshold Signatures with Offline Parties

Alessio Meneghetti

Digital signature schemes are cryptographic primitives used as digital analogous of traditional signatures. These schemes are designed to guarantee several desired properties, among which we recall the three most famous, namely: the non-repudiation, the authentication, and the integrity properties. Together, they guarantee to whomever receives a signed messages that the signer intentionally produced the signature and that the document has not been altered after the signature.
The three components of a signature scheme are:

- a Key-Generation Algorithm, which generates a private key and the corresponding public key,
- a Signing Algorithm, which is the procedure used by the owner of the private key to sign messages,
- a Verification Algorithm, which can be used by anyone who knows the public key to check whether the signature is valid or not.

Observe that the identity of the signer is linked to the private key: only the owner of the private key should be able to produce valid signatures verifiable by using the corresponding public key. According to the context, however, the rightful owner of a message to be signed could be a group of people. Since with classical signature schemes the knowledge of a single private key guarantees the ability to sign, in case of multiple owners there is the need to adopt new solutions capable of distributing in a more flexible manner the right of signing messages.

Threshold Signatures

A (t, n)-threshold signature scheme enables distributed signing among n players, with the rule that enough players, at least t

out of n, can sign messages. Furthermore, no one among the n players can repudiate a valid signature. The first Threshold Multi-Party Signature Scheme was a protocol for producing joint ECDSA signatures proposed by Gennaro, Jarecki, Krawczyk and Rabin [4]. With this first scheme, $t + 1$ players out of $2t + 1$ were required to participate in the signature of messages. Later, MacKenzie and Reiter proposed a second threshold scheme [8]. Further recent improvements by Doerner, Kondi, Lee and Shelat may be found in [3]. The first scheme supporting a general (t, n)-threshold was proposed in 2016 by Gennaro, Goldfeder and Narayanan [5], improved firstly in 2017 by Boneh, Gennaro and Goldfeder [2], and then in 2018 by Gennaro and Goldfeder [6]. An interesting alternative approach by Lindell and Nof has been studied in 2018 [7].
Currently there is a large effort of standardization for threshold signatures, as can be seen in [10].

A key-point in most threshold signature schemes is the compatibility with existing classical signatures algorithms. The idea is to start from a signature scheme and replace the key-generation and signing algorithms with protocols involving all those who possess the right of signing messages, while keeping the verification algorithm identical to that of the centralized algorithm.

More precisely, (t, n)-threshold signature schemes involve n players, each one possessing its own private key established during the multi-party key-generation phase of the protocol. Starting from these n secrets, the players compute a unique public key to be used to verify signatures by using the verification algorithm of a classical signature scheme. When signing, at least t out of the n players initialise a multi-party protocol enabling them to sign messages. To provide an example, in [6] the signatures can be verified by the ECDSA verification algorithm, and any observer is not capable of distinguishing between signatures produced by ECDSA or by the threshold scheme. Notice that all players have to actively participate in the key-generation phase.

Custody Services

A central example in the context of threshold schemes is the rising need to protect users of digital services and owners of digital assets against key loss. In many modern applications (for example, in case of cryptocurrencies) there is no central authority that can restore keys and safely return the ownership of assets to users. Threshold signatures are an elegant solution to this issue: an owner of a digital asset can (partially) distribute the ownership of its assets to multiple parties, to assure resilience against key loss. Services helping users against key loss by partial management of assets or keys are known as custodians. Custody Services are usually composed by three parties:

- the owner of the asset, possessing a personal key;
- the custodian itself, which possesses a second key to be used together with the owner's key in order to sign messages;
- a third party, which does not interact with the user or the custodian unless the owner or the custodian have lost their keys.

In this scenario, the recovery party should be involved only when it is required to recover the wallet funds in case of key loss. In [1] the authors propose a protocol in which the recovery party is involved only once (in a preliminary set-up), and afterwards it is not involved until a lost account must be recovered, i.e. the recovery party does not usually take part in key-generation and signatures phases. The protocol may be seen as an adaption of that in [6]. Its security against (adaptive) adversaries was proved by relying on standard assumptions on the underlying algebraic and geometric problems, such as the strong RSA assumption on semi-prime residue rings and the DDH assumption on elliptic curves. In [9] this scheme has been described, together with some alternatives, as a solution of the custody problem, comparing advantages and disadvantages. With respect to the alternative schemes, the work in [1] enhances practicality by exploiting a party that may stay offline during the key generation.

CrypTOrino 2021
ISBN 000-00-000-0000-0
DOI 10.4399/00000000000009
pag. 37–38 (xx 9999)

Bibliografia

[1] Battagliola, M., Longo, R., Meneghetti, A., and Sala, M., Threshold ECDSA with an Offline Recovery Party, arXiv preprint arXiv:2007.04036, (2020).

[2] Boneh, D., Gennaro, R. and Goldfeder, S., Using level-1 homomorphic encryption to improve threshold dsa signatures for bitcoin wallet security, (2017).

[3] Doerner, J. Kondi, Y., Lee, E., and Shelat, A., Threshold ECDSA from ECDSA assumptions: the multiparty case, 2019 IEEE Symposium on Security and Privacy (SP) (1051-1066), (2019).

[4] Gennaro, R., Jarecki, S., Krawczyk, H, and Rabin, T., Robust threshold DSS signatures, International Conference on the Theory and Applications of Cryptographic Techniques (pp. 354-371), (1997).

[5] Gennaro, R. and Goldfeder, S. and Narayanan, A., Threshold-optimal DSA/ECDSA signatures and an application to Bitcoin wallet security, International Conference on Applied Cryptography and Network Security (156-174), (2016).

[6] Gennaro, R., and Goldfeder, S., Fast multiparty threshold ECDSA with fast trustless setup, 2018 ACM SIGSAC Conference on Computer and Communications Security (1179-1194), (2018).

[7] Lindell, Y. and Nof, A., Fast secure multiparty ECDSA with practical distributed key generation and applications to cryptocurrency custody, 2018 ACM SIGSAC Conference on Computer and Communications Security (1837-1854), (2018)

[8] MacKenzie, P, and Reiter, M. K., Two-party generation of DSA signatures, Annual International Cryptology Conference (pp. 137-154), (2001).

[9] Di Nicola, V., Longo, R., Mazzone, F., and Russo, G., Resilient custody of crypto-assets, and Threshold Multisignatures, Mathematics, 8(10), 1773, (2020).

[10] Brandão, L. T. A. N., Davidson, M., and Vassilev, A., NIST roadmap toward criteria for Threshold Schemes for Cryptographic Primitives, https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8214A.pdf

University of Trento
Alessio.Meneghetti@unitn.it

CrypTOrino 2021
ISBN 000-00-000-0000-0
DOI 10.4399/000000000000010
pag. 39–?? (xx 9999)

Understanding Polynomial Maps over Finite Fields

Austin Dukes, Giacomo Micheli

In this extended abstract we explain how to use algebraic number theory to study polynomial maps over finite fields, which occur everywhere in cryptography and coding theory (APN functions, Reed-Solomon codes, Locally Recoverable Codes). The method in this talk, which is based on techniques from Galois theory and algebraic geometry, has been a particularly useful tool in these areas in recent literature.

Our method was initially used for constructing locally recoverable codes (LRCs) where the known constructions did not work (see [6] for details). In this extended abstract we summarize the context in which our method finds relevance and then briefly describe the method.

The method has also been used to classify functions with low differential uniformity, such as perfect nonlinear functions (PN) and almost perfect nonlinear functions (APN), have been studied extensively (such as in [1] and [5], to name a couple) for their applications in cryptography. Much work has been dedicated to classifying such functions, and nonexistence results for some exceptional monomial PN and APN (and their recent generalizations to PcN and APcN) were obtained in [1] using similar ideas as the ones here.

It is well known that every map from $\mathbb{F}_q$ to $\mathbb{F}_q$ can be written as a polynomial of degree at most $q-1$. Where algebraic number theory most often finds application, however, is when considering a polynomial $f(X) \in \mathbb{F}_q[X]$ of degree n in the regime $n \ll q$. With this context in mind, we now quickly summarize the method to study f as a map.

Let $f \in \mathbb{F}_q[X]$ be a polynomial of degree n. Let $\mathbb{F}_q(t)$ be the field of rational functions in the transcendental t, and denote by M the splitting field of $f(X)-t$ over $\mathbb{F}_q(t)$. Many of the properties of the (polynomial) map $f : \mathbb{F}_q \to \mathbb{F}_q$ are encoded in the Galois group $G := \mathrm{Gal}(M/\mathbb{F}_q(t))$. In particular, analyzing the cycle structures of particular elements of G allows us to obtain asymptotic estimates on the number of $t_0 \in \mathbb{F}_q$ such that $f(X) - t_0$ splits in some desired way.

In what follows, let $F/\mathbb{F}_q$ be a global function field with full constant field $\mathbb{F}_q$, and let M/F be a finite Galois extension with Galois group $G := \mathrm{Gal}(M : F)$. Let $k = \overline{\mathbb{F}}_q \cap M$ be the field of constants of M. Notice that $\mathrm{Gal}(kF : F) \cong \mathrm{Gal}(k : \mathbb{F}_q) \cong C_{[k:\mathbb{F}_q]}$. Let $\gamma \in G$ be such that $\gamma_{|k}$ is the Frobenius automorphism of $k : \mathbb{F}_q$. Let $R \subseteq M$ be a place of M lying above a place $P \subseteq F$ of degree 1. Let $\mathcal{O}_P$ and $\mathcal{O}_R$ be the valuation rings of P and of R, respectively. Write $N := \mathrm{Gal}(M : kF)$. we have $D(R|P)/I(R|P) \cong \mathrm{Gal}(\mathcal{O}_R/R : \mathcal{O}_P/P)$. For any element $\sigma \in N\gamma$ let Γ_σ be the conjugacy class of σ in G. Notice that since G/N is cyclic we have $\Gamma_\sigma \subseteq N\gamma$. Let us recall that if $M : F$ is a Galois extension with Galois group G, and $\sigma \in G$, we say that σ is a Frobenius for P if there exists R lying above P such that $\sigma \in D(R|P)$ and the induced map of σ in $\mathrm{Gal}(\mathcal{O}_R/R : \mathcal{O}_P/P)$ is $x \mapsto x^{q^{\deg(P)}}$.

Let $\mathcal{P}^1(F/\mathbb{F}_q)$ be the set of degree 1 places of a function field $F/\mathbb{F}_q$. We state the following consequence of Chebotarev Density Theorem for the reader's convenience.

Theorem 1. The number of places $P \in \mathcal{P}^1(F/\mathbb{F}_q)$ such that σ is a Frobenius at P is $\frac{|\Gamma_\sigma|}{|N|}(q+1) + O(\sqrt{q})$, where the implied constant can be chosen independently of q.

Interested readers are encouraged to refer to [4, Section II] for more information regarding Theorem 1. We now state the fact which is the key to obtaining the asymptotic estimates mentioned above.

Theorem 2. Let $f(X) \in \mathbb{F}_q[X]$ be a polynomial of degree n. Let ℓ be a positive integer and $d_1, \dots d_\ell$ be positive integers such that $\sum_{i=1}^{\ell} d_i = n$. The number of $t_0 \in \mathbb{F}_q$ such that there exist some distinct irreducible polynomials $p_i \in \mathbb{F}_q[X]$ (depending on t_0) such that $f(X) - t_0 = \displaystyle\prod_{i=1}^{\ell} p_i(X)$ and $\deg p_i = d_i$, is $(|S|/|N|) \cdot q + O(\sqrt{q})$, where S is the subset of elements of N that have cycle decomposition

$$\underbrace{(- - \cdots -)}_{d_1}\underbrace{(- - \cdots -)}_{d_2}\cdots\underbrace{(- - \cdots -)}_{d_\ell}.$$

We temporarily defer the proof of this fact as we wish to first consider some concrete applications.

Let $q = 100003$ and let $f \in \mathbb{F}_q[X]$ be a polynomial of degree 4. Suppose we are interested in finding the number T of elements t_0 in $\mathbb{F}_q$ for which $f(X)$ has exactly four preimages as a map from $\mathbb{F}_q$ to $\mathbb{F}_q$. Notice that this is the same as the number T of elements $t_0 \in \mathbb{F}_q$ such that $f(X) - t_0$ has four zeroes. We first compute the Galois group $G = \mathrm{Gal}(f(X) - t \mid \mathbb{F}_q(t))$ and suppose that the splitting field of $f(X) - t$ has field of constants $\mathbb{F}_q$. For the sake of simplicity of notation (and since it is the generic case), we assume $G = S_4$.

Clearly the number of elements $t_0 \in \mathbb{F}_q$ such that $f(X) - t_0$ has four zeros is the same as the number of t_0 such that $f(X) - t_0 = (X-a)(X-b)(X-c)(X-d)$. Since the identity element of G is the only element with four fixed points, we see that $|S| = 1$. Therefore the fact above gives that the number T of t_0 having 4 preimages is roughly $100003/24 \sim 4166$.

As a more elaborate example, suppose we are in the situation above but are now interesting in finding the number T' of t_0 in $\mathbb{F}_q$ for which $f(X) - t_0$ has exactly two zeroes. We compute $G = \mathrm{Gal}(f(X) - t \mid \mathbb{F}_q(t))$ and verify that the splitting field of $f(X) - t$ has the correct field of constants. Again, we assume $G = S_4$. Note immediately that the number T' is the same as the number of t_0 such that $f(X) - t_0$ factors over $\mathbb{F}_q$ as $f(X) - t_0 = (X-a)(X-b)g(X)$ for some irreducible polynomial g of degree 2. Let $S' = \{(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)\} \subseteq G$ be the set of elements in G fixing exactly 2 points. We see that $|S'| = 6$, so the fact above gives that the number T' of t_0 having exactly 2 preimages is roughly $100003/6 \sim 16667$.

Sketch of the proof of Theorem 2. For simplicity of presentation, we will say that an element $t_0 \in \mathbb{F}_q$ satisfies $(*)$ to mean that the polynomial $f(X) - t_0$ admits a factorization of the form $f(X) - t_0 = \prod_{i=1}^{\ell} p_{t_0,i}(x)$, where the $p_{t_0,i} \in \mathbb{F}_q[X]$ are distinct irreducible polynomials satisfying $\deg p_{t_0,i} = d_i$. Let T be the number of $t_0 \in \mathbb{F}_q$ satisfying $(*)$. By definition T is equal to the number of places $P_{t_0} := \langle t - t_0 \rangle$ such that $f(X) - t_0$ admits the desired factorization. Further $P_{t_0} \in \mathcal{P}^1(F/\mathbb{F}_q)$ for each t_0, where $\mathcal{P}^1(F/\mathbb{F}_q)$ is the set of places of degree 1 of $F := \mathbb{F}_q(t)$. Notice that the t_0 for which $f(X) - t_0$ has a multiple root are only $O(1)$, as they are at most the zeroes of the discriminant $\Delta(t)$ of $f - t$, so we can restrict to unramified places.

Next, fix a place $P := P_{t_0} \subseteq F$ for some unramified $t_0 \in \mathbb{F}_q$ satisfying $(*)$. Let x be a root of $f(X) - t$ in M and let $Q_1, Q_2, \ldots, Q_\ell \subseteq \mathbb{F}_q(x) =: L$ be the places of $\mathbb{F}_q(x)$ lying over P, where Q_i corresponds to the factor $p_{t_0,i}$ of $f(X) - t_0$ and note that $f(Q_i|P) = d_i$. Denote by M the splitting field of $f(X) - t$ over $\mathbb{F}_q(t)$. By Lemma 2.1 of [2] we obtain that a Frobenius for an unramified P can only have cycle decomposition $(d_1, \ldots, d_\ell)$, and vice versa an element of $N\gamma$ with cycle decomposition $(d_1, \ldots, d_\ell)$ will be a Frobenius for an unramified P with the wanted factorization. Moreover, all the Frobeniuses have to lie in $N\gamma$ as they at least have to be the Frobenius for the field of constants extension. We are now ready to use Chebotarev Density Theorem.

Let $S \subseteq N\gamma$ be the elements with orbit decomposition corresponding to the wanted partition $(d_1, \ldots, d_\ell)$. Note that $\Gamma_\sigma \subseteq S$ for every $\sigma \in S$. Then for $\sigma \in S$ we consider the distinct conjugacy classes in S, which form a partition of S. This completes the proof since Theorem 1 can be applied to each conjugacy class, which yields

$$T \sim \frac{1}{|N|}(q+1)\sum_{i=1}^{m} |\Gamma_{\sigma_i}| = \frac{|S|}{|N|}(q+1).$$

□

CrypTOrino 2021
ISBN 000-00-000-0000-0
DOI 10.4399/000000000000011
pag. 43–43 (xx 9999)

Bibliografia

[1] Bartoli, D., Calderini, M., On construction and (non)existence of c-(almost) perfect nonlinear functions (2021), arXiv:2008.03953.

[2] Bartoli, D., Micheli, G., Algebraic constructions of complete m-arcs (2020), arXiv:2007.00911.

[3] Bartoli, D., Micheli, G., Zini, G., Zullo, F., r-fat linearized polynomials over finite fields (2020), arXiv:2012.15357.

[4] Ferraguti, A., Micheli, G., Full classification of permutation rational functions and complete rational functions of degree three over finite fields, Des. Codes Cryptogr. 88, 867–886 (2020).

[5] Mesnager, S., Riera, C., Stănică, P., Yan, H., Zhou, Z., Investigations on c-(almost) perfect nonlinear functions (2021), arXiv:2010.10023v2.

[6] Micheli, G., Constructions of locally recoverable codes which are optimal, IEEE Trans. Inform. Theory 66, 167–175 (2020).

[7] Micheli, G., On the Selection of Polynomials for the DLP Quasi-Polynomial Time Algorithm for Finite Fields of Small Characteristic, SIAM J. Appl. Algebra Geometry 3(2), 256–265 (2019).

[8] Stichtenoth, H., Algebraic Function Fields and Codes (2nd. ed.), Springer Publishing Company, Incorporated (2008).

University of South Florida
gmicheli@usf.edu

CrypTOrino 2021
ISBN 000-00-000-0000-0
DOI 10.4399/000000000000012
pag. 45–?? (xx 9999)

A multifactor RSA-like scheme

Emanuele Bellini, Nadir Murru

In the RSA scheme, the decryption procedure is the most time–consuming task, since the encryption exponent is usually taken with an efficient binary representation, on the contrary the decryption exponent has not this property. Thus, some variants of the RSA scheme, using a modulus with a multi-factor modulus, i.e. a product of more than two primes or powers of primes, have been proposed in order to improve its efficiency [3,5,6,9]. Moreover, there exists RSA–like schemes that provide a decryption procedure faster than RSA and also more security in broadcast scenarios [7]. These schemes are usually based on isomorphisms between two groups, one of which is the set of points over a cubic or a conic. In [4], a generalization of the KMOV cryptosystem [8] has been proposed, thus generalizing a RSA-like scheme based on elliptic curves and a modulus $N = pq$ to a similar scheme based on the generic modulus $N = p^r q^s$. Here, we present a similar generalization of the scheme [1], which is based on the Pell's conic, to the modulus $N = p_1^{e_1} \cdot \ldots \cdot p_r^{e_r}$ for $r > 2$, for further details see [2]. The Pell's conic, over a field $\mathbb{K}$ is defined by

$$\mathcal{C} = \{(x, y) \in \mathbb{K}^2 : x^2 - Dy^2 = 1\}$$

for a given $D \in \mathbb{K}$, which is not a square in our application. If we consider the following product

$$(x_1, y_1) \otimes (x_2, y_2) = (x_1x_2 + Dy_1y_2, x_1y_2 + x_2y_1),$$

then $(\mathcal{C}, \otimes)$ is an abelian group whose identity is $(1, 0)$. We highlight that the group structure on $\mathcal{C}$ is obtained using only the affine points. Moreover, there exists explicit isomorphisms between $(\mathcal{C}, \otimes)$ and $(\mathbb{K} \cup \alpha, \odot)$, where α is not an element of $\mathbb{K}$ and it is the identity. The product is then defined by

$$a \odot b = \frac{ab + D}{a + b},$$

if $a + b \neq 0$; $a \odot b = \alpha$, if $a + b = 0$, that is a is the inverse of b and viceversa. For the cryptographic applications, it is important to observe that when $\mathbb{K} = \mathbb{Z}_p$, then $\mathcal{C}$ is a cyclic group of order $p + 1$. Moreover, if we consider the Pell's conic with points that belong to the ring $\mathbb{Z}_N$, with $N = pq$, for p, q primes, then

$$(x, y)^{(p+1)(q+1)} = (1, 0)$$

for every $(x, y) \in \mathcal{C}$, with $y \in \mathbb{Z}_N^*$. Exploting the above considerations, it is possible to construct an RSA–like scheme having decryption operation two times faster than RSA, and involving the lowest number of modular inversions with respect to other RSA–like schemes based on curves. For generalizing such scheme with multifactor moduli, first of all it is necessary to study the behaviour of the Pell's conic when the points lie over a ring $\mathbb{Z}_{p^r}$. In particular we recall from [2] the following theorem.

Theorem 1. The Pell's conic

$$\mathcal{C} = \{(x, y) \in \mathbb{Z}_{p^r} \times \mathbb{Z}_{p^r} : x^2 - Dy^2 = 1\}$$

with the product $\otimes$ is a cyclic group of order $p^{r-1}(p + 1)$, for D non–quadratic residue in $\mathbb{Z}_{p^r}$.

Then, the multifactor RSA–like scheme works in the following way.

Key generation. The key generation is performed by the following steps:

— choose r prime numbers $p_1, \ldots, p_r$, r odd integers $e_1, \ldots, e_r$ and compute $N = \prod_{i=1}^{r} p_i^{e_i}$;

— choose an integer e such that $gcd(e, lcm \prod_{i=1}^{r} p_i^{e_i-1}(p_i + 1)) = 1$;

— evaluate $d = e^{-1} \pmod{lcm \prod_{i=1}^{r} p_i^{e_i-1}(p_i + 1)}$.

The public or encryption key is given by (N, e) and the secret or decryption key is given by $(p_1, \ldots, p_r, d)$. Encryption. We can encrypt pairs of messages $(M_x, M_y) \in \mathbb{Z}_N^* \times \mathbb{Z}_N^*$, such that $\left(\dfrac{M_x^2 - 1}{N}\right) = -1$. This condition will ensure that we can perform all the operations. The encryption of the messages is performed by the following steps:

— compute $D = \dfrac{M_x^2 - 1}{M_y^2} \pmod{N}$

— compute $M = \dfrac{M_x + 1}{M_y} \pmod{N}$;
— compute the ciphertext $C = M^{\odot e} \pmod{N}$.

Notice that not only C, but the pair (C, D) must be sent through the insecure channel. Decryption The decryption is performed by the following steps:

— compute $C^{\odot d} \pmod{N}$, which will be equal to M;
— compute $\left(\dfrac{M^2 + D}{M^2 - D}, \dfrac{2M}{M^2 - D}\right) \pmod{N}$ for retrieving the messages (M_x, M_y).

Let us observe that our scheme encrypts and decrypts messages of size $2 \log N$. To decrypt a ciphertext of size $2 \log N$ using CRT, standard RSA requires four full exponentiation modulo $N/2$-bit primes. Basic algorithms to compute $x^d \mod p$ requires $O(\log d \log^2 p)$, which is equal to $O(\log^3 p)$ if $d \sim p$.
Using CRT, if $N = p_1^{e_1} \cdot \ldots \cdot p_r^{e_r}$, our scheme requires at most r exponentiation modulo N/r-bit primes.
This means that the final speed up of our scheme with respect to RSA is

$$\frac{4 \cdot (N/2)^3}{r \cdot (N/r)^3} = r^2/2. \tag{1}$$

When $r = 2$ our scheme is two times faster than RSA, as it has already been shown in [1]. If $r = 3$ our scheme is 4.5 time faster, with $r = 4$ is 8 times faster, and with $r = 5$ is 12.5 times faster.
The proposed scheme offers the same security as RSA in a one-to-one communication scenario and is more secure in broadcast applications. Indeed, in this case we have that a plaintext is encrypted using the same public exponent and different moduli. It is well-known that in this scenario it is possible to recover the plaintext by solving a set of congruences of polynomials [7]. However, this attack fails when the trapdoor function is not a simple monomial power as in RSA. This allows the use of smaller public exponents even in broadcast scenarios.

CrypTOrino 2021
ISBN 000-00-000-0000-0
DOI 10.4399/000000000000013
pag. 49–49 (xx 9999)

Bibliografia

[1] Bellini, E., Murru, N., An efficient and secure RSA-like cryptosystem exploiting R ́edei rational functions over conics, Finite Fields Appl. 39, 179–194 (2016).

[2] Bellini, E., Murru, N., A Multi-factor RSA-like Scheme with Fast Decryption Based on Rédei Rational Functions over the Pell Hyperbola, NUMTA 2019, LNCS 11973, 1–15 (2020).

[3] Boneh, D., Shacham, H., Fast variants of RSA, CryptoBytes 5(1), 1–9 (2002).

[4] Boudabra, M., Nitaj, A., A new generalization of the KMOV cryptosystem, J. Appl. Math. Comput. 57, 1–17 (2017).

[5] Ciet, M., Koeune, F., Laguillaumie, F., Quisquater, J., Short private exponent attacks on fast variants of RSA, UCL Crypto Group Technical Report Series CG-2003/4, Université Catholique de Louvain (2002).

[6] Compaq, Cryptography using Compaq multiprime technology in a parallel processing environment, ftp://15.217.49.193/pub/solutions/CompaqMultiPrimeWP.pdf, Accessed 2021.

[7] Hastad, J., using RSA with low exponent in a public key network, Williams, H.C. (ed.) CRYPTO 1985. LNCS 218, 403–408 (1986).

[8] Koyama, K., Maurer, U.M., Okamoto, T., Vanstone, S.A., New public-key schemes based on elliptic curves over the Ring $\mathbb{Z}_n$, Feigenbaum, J. (ed.) CRYPTO 1991, LNCS 576, 252–266 (1992).

[9] Takagi, T., Fast RSA-type cryptosystem modulo p^k, In: Krawczyk, H. (ed.) CRYPTO 1998, LNCS 1462, Springer, Heidelberg, 318–326 (1998).

University of Trento
nadir.murru@unitn.it

CrypTOrino 2021
ISBN 000-00-000-0000-0
DOI 10.4399/000000000000014
pag. 51–?? (xx 9999)

Privacy-preserving signatures from isogenies

Federico Pintore

1. Introduction

Ring signatures, introduced by Ron Rivest, Adi Shamir and Yael Tauman [7], allow a signer to produce a signature on behalf of a group of users, called ring, while the signature hides the real identity of the signer. Recently, they have found application in electronic voting systems, and in a cryptocurrency which provides privacy by default, named Monero.

Unfortunately, the security of the ring signatures used in Monero relies on the hardness of the discrete logarithm problem, which can be solved in quantum polynomial time [8]. Given the recent significant advancements in quantum computing, there is the need to determine quantum-resistant replacements.

Efficient ring signatures which are secure even in the presence of quantum computers have been proposed both from lattice-based and symmetric-key assumptions [6, 21]. Among the areas of post-quantum cryptography, isogeny-based cryptography is a relatively new field which is particularly appealing due to the small key sizes required (we refer, for example, to the isogeny-based signature schemes [2, 3]).

In [1], Ward Beullens, Shuichi Katsumata and the author of this work proposed the first efficient isogeny-based ring signature scheme, named Calamari. The size of the produced signatures is logarithmic in N (where N is the number of signers, i.e. the cardinality of the ring), and it is an order of magnitude smaller than all previously-known post-quantum ring signatures. The dependence on N of the size of the signatures generated with Calamari is due to a small number of paths in Merkle trees of depth $\log N$. Previous works rely on hidden paths in Merkle trees, whose consistency is proved in zero-knowledge, and in that cases the multiplicative factor of $\log N$

is much larger.

In the following, the main building block of Calamari is briefly described.

2. Sigma Protocols, OR-proofs and Ring Signatures

Let X and Y be two finite sets, and $\mathcal{R} \subset X \times Y$ a polynomially-computable binary relation, i.e. given $(\mathsf{X}, \mathsf{W}) \in X \times Y$, it can be checked in time $\mathsf{poly}(|\mathsf{X}|)$ whether (X, W) belongs to $\mathcal{R}$ or not. A Sigma protocol for $\mathcal{R}$ is an interactive protocol between a prover and a verifier, composed by a challenge set Ch, whose cardinality is exponential in a security parameter λ, and four algorithms: P_1 and P_2 run by the prover, and V_1 and V_2 run by the verifier.

The prover holds a pair $(\mathsf{X}, \mathsf{W}) \in \mathcal{R}$, while the verifier holds only X. The goal of the interaction is making the prover convince the verifier that they possess X, without revealing anything about X itself. The interaction starts with the prover, who produces a commitment running P_1; the verifier runs V_1 to produce a uniform challenge ch; the prover runs P_2, obtaining a response; finally, the verifier runs V_2 to either accept or reject the response.

The standard security properties required to a Sigma protocol are correctness (if both actors honestly follow the protocol, the verifier must always accept), special soundness (which prevents a cheating prover from convincing the verifier) and honest-verifier zero-knowledge (which assures the interaction does not leak information about W).

An OR-Proof for the binary relation $\mathcal{R}$ is a Sigma protocol for the extended binary relation $\mathcal{R}_{\text{OR}}$ defined as:

$$\mathcal{R}_{\text{OR}} = \{((\mathsf{X}_1, \ldots, \mathsf{X}_N), (\mathsf{W}, I)) | N \in \mathbb{N}^*, I \in [N],\ (\mathsf{X}_I, \mathsf{W}) \in \mathcal{R}\}$$

where $[N] = \{1, \ldots, N\}$. By applying the Fiat-Shamir transform [4], an OR-Proof can be turned into a ring signature. In particular, such transform replaces V_1 with a Random Oracle (an idealised random function, which is instantiated with a hash function in practice) and a ring signature is a tuple composed by the commitment, the challenge and the response. In terms of efficiency, the compactness of the ring signatures produced with a transformed OR-Proof depends on the response size of the OR-Proof.

3. A new OR-Proof

Calamari is the ring signature obtained by transforming a new OR-Proof for group actions satisfying some cryptographic properties [1, Def. 8]. Below, such OR-Proof is described by considering a free and transitive action $\star$ of a cyclic finite group $\mathbb{G}$, having order ℓ and generator $\mathfrak{g}$, on a set S of supersingular elliptic curves defined over a prime field $\mathbb{F}_p$. This is an isogeny-based action since the action of a group element on an elliptic curve E corresponds to an isogeny from E (an isogeny is a morphism which sends the zero element into the zero element).

Consider $\mathsf{X} = (\mathsf{X}_1, \ldots, \mathsf{X}_N) \in S^N$, with $\mathsf{X}_i = \mathfrak{g}^{s_i} \star E_0$ for $i \in [N]$. $E_0 \in S$ is a fixed curve, while $s_i \in \mathbb{Z}_\ell$. Let $\mathsf{W} = s_I$ for some $I \in [N]$.

The algorithm P_1 uniformly samples α from $\{0,1\}^\lambda$ and uses it as input of a second Random Oracle PRG. The corresponding output consists of $r \in \mathbb{Z}_\ell$ and $\mathsf{bits}_i \in \{0,1\}^\lambda$, for $i \in [N]$. Then, it computes $\mathsf{com}_i = \mathsf{Com}(\mathfrak{g}^{s_i+r} \star E_0, \mathsf{bits}_i)$ by using a third Random Oracle Com, and produces a index-hiding Merkle tree [1, Sec. 2.6] from the leaves $\mathsf{com}_1, \ldots, \mathsf{com}_N$. The root of the tree, denoted by root, is the commitment com output by P_1. V_1 samples a uniform bit ch as challenge. P_2 outputs $s_I + r$, bits_I and the path path which connects com_I to root when $\mathsf{ch} = 0$, and α when $\mathsf{ch} = 1$. V_2 reconstructs the root of the tree when $\mathsf{ch} = 1$, and compares it to root. When $\mathsf{ch} = 0$, V_2 computes $\mathsf{Com}(\mathfrak{g}^{s_I+r} \star E_0, \mathsf{bits}_I)$ and verifies this is a leaf of the tree with root root by using path. If the checks are positive, V_2 accepts the response. The described interaction is repeated in parallel λ-times to increase the size of the challenge space.

We conclude by observing that only the size of path depends (logarithmically) on N. Moreover, in [1, Sec. 3.4] some optimisations are introduced to further decrease the size of the ring signatures.

CrypTOrino 2021
ISBN 000-00-000-0000-0
DOI 10.4399/000000000000015
pag. 55–55 (xx 9999)

Bibliografia

[1] Beullens, W., Katsumata, S., Pintore, F., Calamari and Falafl: Logarithmic (linkable) ring signatures from isogenies and lattices, International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT) 2020, pp. 464–492.

[2] Beullens, W., Kleinjung, T., Vercauteren, F., CSI-FiSh: Efficient isogeny based signatures through class group computations, International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT) 2019, pp. 227–247.

[3] El Kaafarani, A., Katsumata, S., Pintore, F., Lossy CSI-FiSh: Efficient signature scheme with tight reduction to decisional CSIDH-512, IACR International Conference on Public-Key Cryptography (PKC), 2020, pp. 157–186.

[4] Fiat, A., Shamir, A., How to prove yourself: Practical solutions to identification and signature problems, Conference on the theory and application of cryptographic techniques (CRYPTO) 1986, pp. 186–194.

[5] Katz, J., Kolesnikov, V., Wang, X., Improved non-interactive zero knowledge with applications to post-quantum signatures, Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 525-537.

[6] Lu, X., Au, M. H., Zhang, Z., Raptor: a practical lattice-based (linkable) ring signature, International Conference on Applied Cryptography and Network Security, 2019, pp. 110–130.

[7] Rivest, R.L., Shamir, A., Tauman, Y., How to leak a secret, International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT) 2001, pp. 552–565.

[8] Shor, P. W. Algorithms for quantum computation: discrete logarithms and factoring, Proceedings of the 35th Annual Symposium on Foundations of Computer science (Santa Fe, NM, 1994), pp. 124–134. IEEE.

University of Bari
federico.pintore@uniba.it

CrypTOrino 2021
ISBN 000-00-000-0000-0
DOI 10.4399/000000000000016
pag. 57–?? (xx 9999)

The integer factorization problem in cryptography

Giordano Santilli

1. Introduction

One of the first formulations of the Fundamental Theorem of Arithmetic can be found in Euclid's Elements. The modern statement is the following:

Theorem 1.1. Every positive integer $N \geq 2$ can be factored in a unique way as the product of prime powers, i. e. there exists a positive integer s such that

$$N = p_1^{\alpha_1} \cdots p_s^{\alpha_s},$$

where $p_1, \dots, p_s \in \mathbb{N}^+$ are the prime factors and $\alpha_1, \dots, \alpha_s \in \mathbb{N}^+$ are the corresponding exponents.

The Fundamental Theorem of Arithmetic enable us to define the Integer Factorization Problem (IFP): given a composite number N find its unique factorization. Often, instead of considering a generic composite number, it is more convenient to restrict to a semiprime, i. e. $N = pq$, with p and q prime numbers. At the moment, it does not exist any classical algorithm that solves the IFP in polynomial time.

In this abstract some cryptographic schemes which rely on the IFP will be presented, as well as some factorization algorithms used to solve the IFP. Finally, I present an interesting property on remainders which given an integer N and under suitable conditions, allows to obtain a formula to describe all the remainders of N with respect to a sequence of consecutive integers as a quadratic modular relation. This work comes from an idea of M. Sala and M. Piva and has been developed in my Ph.D. thesis [11].

2. Cryptographic protocols

As usual in public-key cryptosystem, one-way mathematical problems, as the IFP, are used as fundamental bricks for the creation of new schemes. The most famous one is surely RSA [10], in which two primes of almost the same length p and q are generated and multiplied together to obtain $N = pq$. Moreover we need two integers $d, e \in \mathbb{Z}_N$ such that $de \equiv 1 \bmod \varphi(N)$. Alice must keep the values of p, q and d secret, corresponding to the private key, while N and e are known by anyone and they form the public key. In order to send a message $m < N$ to Alice, Bob needs to compute $c \equiv m^e \bmod N$, then he sends it to Alice, who may decrypt it by computing $c^d \equiv (m^e)^d \equiv m \bmod N$. The security of this protocol in mainly based on the IFP: it can be proved that finding $\varphi(N)$ from N or computing the inverse of $e \bmod \varphi(N)$ are as difficult as finding the primes p and q from N, i. e. finding the factorization of N.

Other examples of cryptographic schemes based on the IFP are: Rabin cryptosytem [9] and Goldwasser-Micali cryptosystem [1], which employ quadratic residues modulo a semiprime; Paillier cryptosystem [6], an additive homomorphic scheme used in many applications; Okamoto-Uchiyama cryptosystem [5], which considers instead integers modulo p^2q, with p and q prime numbers.

3. Factorization methods

The most natural factorization method is the so-called Trial Division i. e. trying to divide N for any prime smaller than $\sqrt{N}$. Clearly, this method is fast when N has a small factor. This characteristic may be used to distinguish factorization techniques into two categories: algorithms which try to recover the smaller factor of N; algorithms which do not take into account the size of the factors of N. The first kind of algorithms are more effective when one of the prime factors of N has between 7 and 40 digits. Apart from Trial Division other notable examples of this type of algorithms are Pollard's ρ [7] or Lenstra's Elliptic Curves Method (ECM) [2]. The second kind of algorithms is used when N has more than 100 digits and no small factors. Famous examples of methods of this kind are the Continued Fraction Method (CFRAC) [4], the Quadratic Sieve [8] and the General Number Field Sieve (GNFS) [3]. As said before, none of this methods works in polynomial time and, at the moment, the best performing method is GNFS, which has a subexponential complexity. It is interesting to note that many built-in factoring functions in several programs employ methods of both categories. For example the `Factorization` command in MAGMA, after a brief search in a database, performs a trial division up to 10000, then tries with Pollard's ρ. If the result is still not found, it applies other two algorithms: ECM and Multiple Polynomial Quadratic Sieve (MPQS).

4. Successive remainders

In [11], we have noticed an interesting properties of integers modulo a semiprime N. Let $m \in \mathbb{N}^+$ be such that $\left\lfloor \sqrt{\frac{N}{2}} \right\rfloor + 1 \leq m \leq \left\lfloor \sqrt{N} \right\rfloor - 1$, and compute

$$\begin{cases} N \equiv a_0 \bmod m \\ N \equiv a_1 \bmod (m+1) \\ N \equiv a_2 \bmod (m+2). \end{cases}$$

If the sequence $\{a_0, a_1, a_2\}$ is monotonic, i. e. $a_0 \leq a_1 \leq a_2$ or $a_0 \geq a_1 \geq a_2$ it is possible to compute the interpolating polynomial f of degree 2 such that $f(i) = a_i$ for $i = 1, 2, 3$. In this setting the following proposition holds:

Proposition 4.1. The polynomial $f(x) \in \mathbb{Q}[x]$ defined as above is such that

$$N \equiv f(i) \bmod (m+i),$$

for any $i \in \mathbb{Z}$.

This means that the sequence of successive remainders of a given semiprime N can be predicted by a quadratic polynomial. Clearly if it is possible to find an $i \in \mathbb{Z}$ such that $f(i) \equiv 0 \bmod (m+i)$, then $m+i$ is a factor of N. However, it can be shown that recovering such an i is a problem equivalent to the IFP problem. Still, if the starting modulo m is chosen close to the factor p, then the polynomial f has at least a zero in $\mathbb{Z}$, which enables us to recover a factor of N.

CrypTOrino 2021
ISBN 000-00-000-0000-0
DOI 10.4399/000000000000017
pag. 61–61 (xx 9999)

Bibliografia

[1] Goldwasser, S., Micali, S., Probabilistic encryption, Journal of Computer and System Sciences, 28(2), 270-299, (1984).

[2] Lenstra Jr., H. W., Factoring integers with elliptic curves, Annals of Mathematics, 649-673, (1987).

[3] Lenstra, A. K., Lenstra, H. W., Manasse, M. S., Pollard, J. M., The number field sieve in The development of the number field sieve (pp. 11-42). Springer, Berlin, Heidelberg, (1993).

[4] Lehmer, D. H., Powers, R. E., On factoring large numbers, Bulletin of the American Mathematical Society, 37(10), 770-776, (1931).

[5] Okamoto, T., Uchiyama, S., A new public-key cryptosystem as secure as factoring, In International Conference on the Theory and Applications of Cryptographic Techniques (pp. 308-318), Springer, Berlin, Heidelberg, (1998).

[6] Paillier, P., Public-key cryptosystems based on composite degree residuosity classes, in International Conference on the Theory and Applications of Cryptographic Techniques (pp. 223-238), Springer, Berlin, Heidelberg, (1999).

[7] Pollard, J. M., A Monte Carlo method for factorization. BIT Numerical Mathematics, 15(3), 331-334, (1975).

[8] Pomerance, C., Analysis and comparison of some integer factoring algorithms, Computational Methods in Number Theory, 89-139, (1982).

[9] Rabin, M. O., Digitalized signatures and public-key functions as intractable as factorization, Massachusetts Inst. of Tech. Cambridge Lab. for Computer Science, (1979).

[10] Rivest, R. L., Shamir, A., Adleman, L., A method for obtaining digital signatures and public-key cryptosystems, Communications of the ACM, 21(2), 120-126, (1978).

[11] Santilli, G., An investigation on Integer Factorization applied to Public Key Cryptography, Doctoral dissertation, University of Trento, (2019).

University of Trento
giordano.santilli@unitn.it

CrypTOrino 2021
ISBN 000-00-000-0000-0
DOI 10.4399/000000000000018
pag. 63–?? (xx 9999)

On the classical authentication in Quantum Key Distribution

Guglielmo Morgari, Edoardo Signorini, Francesco Stocco

1. Introduction

Quantum Key Distribution (QKD) protocols require the presence of an authenticated classical channel in order to guarantee overall correctness and security [1]. In the private-key scenario, the main tools used to build an authenticated channel are the Message Authentication Codes (MAC). Alice and Bob share a secret key k before initiating the communication and use it to authenticate messages exchanged over the insecure channel. At a high level, if Alice wants to send an authenticated message m to Bob, she uses a public function in combination with the key k and the message m to derive a tag t, then sends the pair (m, t) to Bob. Bob repeats the same process to derive the tag t and compares it with the one received to verify that the message has not been forged or modified. The authentication scheme is considered secure if, without the knowledge of the key k, an attacker has a negligible probability of successfully forging authenticated messages, even after intercepting valid pairs (m, t). Since QKD wants to build a key exchange protocol with unconditional security, the use of Information-Theoretically Secure (ITS) authentication protocols is required [2], for which an attacker should have no better strategy than randomly choosing a tag, regardless of its computing power. Moreover, in the context of practical QKD, it is crucial to study what happens when authentication schemes are used with partially known keys. After a brief overview of the underlying theory, this will be the main topic of this abstract.

2. Technical background

We will use the following notation. Let (Ω, P) be a discrete probability space and $X : \Omega \to \mathcal{X}$ be a random variable, P_X denotes the probability distribution of X. Given another random variable over the same sample space $Y\colon \Omega \to \mathcal{Y}$, the conditional probability distribution of X given $Y = y$ is denoted with $P_{X|Y=y}$. For the random variable X the Shannon entropy is defined as $H(X) = -\sum_{x\in\mathcal{X}} P_X(x)\log P_X(x)$. The statistical distance between probability distributions P_X and P_Y, over the same alphabet $\mathcal{X}$, is $\delta(P_X, P_Y) = \frac{1}{2}\sum_{x\in\mathcal{X}}|P_X(x) - P_Y(x)|$. $\mathcal{M}, \mathcal{T}$ and $\mathcal{K}$ are fixed-length binary strings sets denoting the message space, the tag space and the key space respectively. A key $k \in \mathcal{K}$ is said to be ε-perfect if its distribution has statistical distance ε from the uniform distribution over $\mathcal{K}$.

ITS authentication schemes can be built from a particular class of hash functions, called ε-Almost Strongly Universal$_2$ (ε-ASU$_2$), which were introduced by Wegman and Carter in their seminal work [3, 4] and later formalized by Stinson [5]. In the following, we will focus on these constructions since they are widely used in theoretical QKD analysis and in commercial QKD systems [6, 7].

Let $\{h_k\colon \mathcal{M} \to \mathcal{T}\}_{k\in\mathcal{K}}$ be a family of ε-ASU$_2$ functions, then the following condition is satisfied: given any $m_1, m_2 \in \mathcal{M}, m_1 \neq m_2$ and $t_1, t_2 \in \mathcal{T}$ it holds that $|\{k \in \mathcal{K} \mid h_k(m_1) = t_1\}| = |\mathcal{K}|/|\mathcal{T}|$ and $|\{k \in \mathcal{K} \mid h_k(m_1) = t_1, h_k(m_2) = t_2\}| \leq \varepsilon|\mathcal{K}|/|\mathcal{T}|$. In other words the digests $h_k(m_1), h_k(m_2)$ are independently and almost uniformly distributed in the tag space. Given this condition and assuming that the key k is uniformly chosen for each authentication round, it is easy to show that the probability of an attacker to forge a valid message-tag pair (m', t') is initially $1/|\mathcal{T}|$ and is at most ε after intercepting a valid pair (m, t).

The full-fledged security of a cryptographic protocol does not imply that it can be arbitrarily combined with other protocols while maintaining the same level of security. The Universal Composability (UC) framework [8] is a strong model in which protocols (resources) are proven to be secure even when arbitrarily combined with each other. Security is formulated in terms of indistinguishability, meaning that a distinguisher should not be able to determine whether it is interacting with the real implementation or the ideal functionality, except at most with probability ε. Such a resource is called ε-UC-secure.

The main result linking authentication schemes based on ε-ASU$_2$ hash functions and the UC framework is due to Abidin and Larsson. In [9] they proved that using such a scheme with an ε'-perfect key is $\varepsilon + \varepsilon'$-UC-secure. This result implies that the schemes proposed by Wegman and Carter can be composed with QKD protocols, which are proved UC-secure and produce ε'-perfect keys [10].

3. Key recycling with partially known key

In their work, Wegman and Carter proposed also an authentication scheme able to recycle a portion of the already used key. Their proposal is built from a family of ε-ASU$_2$ hash functions $\{h_{k_1} : \mathcal{M} \to \mathcal{T}\}_{k_1 \in \mathcal{K}}$ and the authentication proceeds as follows: Alice and Bob share a key $(k_1, k_2) \in \mathcal{K} \times \mathcal{T}$, the tag for the message $m \in M$ is produced as $t = h_{k_1}(m) \oplus k_2$. In subsequent authentication rounds, k_1 is kept fixed and only a new OTP (One-time pad) key k_2 needs to be exchanged. If k_2 is uniformly chosen in $\mathcal{T}$ then k_2 hides h_{k_1} and it is easy to prove that $\{g_{k_1,k_2}(\cdot) := h_{k_1}(\cdot) \oplus k_2 \mid (k_1, k_2) \in \mathcal{K} \times \mathcal{T}\}$ is a family of ε-ASU$_2$ hash functions.

The required key length of this scheme approximates $\log|\mathcal{T}|$ per round and is therefore optimal. It is of particular interest in combined use with QKD, where a portion of the generated key is used as the authentication key in the next round, and the length of the latter has a direct impact on the overall key rate. In [11], Portmann has proved that, when using uniformly random keys, this authentication scheme is UC-composable and is therefore usable with QKD protocols.

4. Our contribution

In our ongoing work, we propose to extend the result of [9] to the key-recycling authentication scheme, analyzing the case in which the key k_2 is not uniformly chosen in $\mathcal{T}$, as occurs when it comes from an imperfect resource such as QKD. We evaluate the security of the authentication scheme and estimate the increase of the attacker information on k_1, which is propagated in subsequent authentication rounds. Let K_H, K_X be two independent random variables modelling uniformly chosen keys over $\mathcal{K}, \mathcal{T}$. Let Z be a random variable modelling the information an attacker might have about the keys K_H, K_X. Suppose the attacker has seen a value $Z = z$ such that $H(K_H \mid Z = z) \geq \log(|\mathcal{K}|) - \varepsilon_H^2$ and $H(K_X \mid Z = z) \geq \log(|\mathcal{T}|) - \varepsilon_X^2$. Then the authentication scheme based on the ε-ASU$_2$ hash functions $\{h_{K_H}(\cdot) \oplus K_X\}$, is $\varepsilon + \sqrt{\frac{\ln(2)}{2}}(\varepsilon_H + \varepsilon_X)$-UC-secure. Moreover after revealing a couple (m, t), with $t = h_{K_H}(m) \oplus K_X$, the information of the attacker on K_H is expected to increase at most by ε_X^2.

CrypTOrino 2021
ISBN 000-00-000-0000-0
DOI 10.4399/000000000000019
pag. 67–68 (xx 9999)

Bibliografia

[1] Scarani, V., Bechmann-Pasquinucci, H., Cerf, N.J., Dušek, M., Lütkenhaus, N. and Peev, M. The security of practical quantum key distribution, Reviews of Modern Physics, vol. 81, pp. 1301–1350, (2009).

[2] Pacher, C., Abidin, A., Lorünser, T., Peev, M., Ursin, R., Zeilinger, A. and Larsson, J.Å., Attacks on quantum key distribution protocols that employ non-ITS authentication, Quantum Information Processing, vol. 15, pp. 327–362, (2016).

[3] Carter, J. and Wegman, M. N., Universal classes of hash functions, Journal of Computer and System Sciences, vol. 18, pp. 143–154, (1979).

[4] Wegman, M. N. and J. Carter, New hash functions and their use in authentication and set equality, Journal of Computer and System Sciences, vol. 22, pp. 265–279, (1981).

[5] Stinson, D. R., Universal hashing and authentication codes, Designs, Codes and Cryptography, vol. 4, pp. 369–380, (1994).

[6] Sasaki, M. et. al, Field test of quantum key distribution in the Tokyo QKD Network, Optics Express, vol. 19, p. 10387, (2011).

[7] Walenta, N. et. al, A fast and versatile quantum key distribution system with hardware key distillation and wavelength multiplexing, New Journal of Physics, vol. 16, p. 013047, (2014).

[8] Canetti, R. Universally composable security: A new paradigm for cryptographic protocols, in Proceedings 42nd IEEE Symposium on Foundations of Computer Science, (Newport Beach, CA, USA), pp. 136–145, IEEE, (2001).

[9] Abidin, A. and Larsson, J.-Å., Direct proof of security of Wegman–Carter authentication with partially known key, Quantum Information Processing, vol. 13, pp. 2155–2170, (2014).

[10] Müller-Quade, J. and Renner, R., Composability in quantum cryptography, New Journal of Physics, vol. 11, p. 085006, (2009).

[11] Portmann, C., Key Recycling in Authentication, IEEE Transactions on Information Theory, vol. 60, pp. 4383–4396, (2014).

Telsy S.p.A.
edoardo.signorini@telsy.it

Parte II
INVITED SURVEYS

CrypTOrino 2021
ISBN 000-00-000-0000-0
DOI 10.4399/000000000000020
pag. 71–?? (xx 9999)

A survey on NIST PQ signatures

Nicola Di Chiano, Riccardo Longo, Alessio Meneghetti, Giordano Santilli

Shor's shockingly fast quantum algorithm for solving the period-finding problem is a threat for the most common public-key primitives, as it can be efficiently applied to solve both the Integer Factorisation Problem and the Discrete Logarithm Problem. In other words, as soon as a large-enough quantum computer is born, many once-secure protocols have to be replaced by still-secure alternatives. Instead of relying, for example, on the RSA protocol, the Diffie-Hellman key-exchange or the (Elliptic Curve) Digital Signature Algorithm, many researchers moved their attention to the design and analysis of primitives which are yet to be broken by quantum algorithms.
The urgency of the threat imposed by quantum computers led the U.S. National Institute of Standards and Technology (NIST) to open calls for both Post-Quantum Public-Keys Exchange Algorithms and Post-Quantum Digital Signature Algorithms [32]. This new NIST standardisation process started in 2016, has involved hundreds of researchers, has seen 37 early submissions for a total of 82 proposals, and has recently reached its third round of analyses.
In this brief survey we focus on the round 3 finalists and alternate candidates for Digital Signatures, announced on July 22, 2020:

Finalists	Alternate Candidates
CRYSTALS-DILITHIUM [4]	SPHINCS$^+$ [3]
FALCON [16]	GeMSS [10]
Rainbow [13]	Picnic [11]

These schemes are designed to address distinct security levels, known as Security Level I, III and V. These levels correspond to, respectively, 128, 192 and 256 bits of security against collisions. Among the six schemes above, only Falcon cannot be instantiated to all three security levels. In order to present these primitives we start, in Section 1, with an introduction to their underlying mathematical objects and the related problems, i.e. lattices, polynomial ideals, one-way functions and zero-knowledge proofs. Then, Section 2 describes the six digital signatures and lists their algorithms for key-generation, signing and verification. Finally, in Section 3 we conclude with a comparison between the different schemes.

1. Preliminaries

1.1. Digital Signatures

A digital signature is a public-key protocol that acts as the digital counterpart of a traditional signature. Formally, the properties that a digital signature must achieve are the following [22]:

1) Authentication: the receiver of the document must be sure of the identity of the sender.
2) Integrity: the signed document should not be altered when transmitted.
3) Non-repudiation: the signer of the document cannot deny having signed the document.
4) Non-reusability: the signature must be used only once.
5) Unforgeability: only the signer of the message should be able to give a valid signature.

A signature scheme is usually composed by three algorithms:

- Keys Generation Algorithm: using the global parameters defined at the beginning of the scheme, this algorithm generates a private key and the corresponding public key.
- Signing Algorithm: using the private key and the message needed to be signed, this algorithm outputs a signature.
- Verification Algorithm: using the public key and the message, the receiver is able to decide whether the signature obtained is valid or not.

A detailed description of digital signature schemes can be found in [29, 41].

1.2. Lattice Theory

Definition 2. Let m be a positive integer. A discrete additive subgroup L of $\mathbb{R}^m$ is called a (Euclidean) lattice. An equivalent definition may be given in terms of linear algebra: given a finite set of linearly independent vectors $B = \{v_1, \ldots, v_n\}$ of $\mathbb{R}^m$, a lattice $L(B)$ is the set of the linear combinations with integer coefficients of the set B. The set B is called a basis of the lattice L and n is the dimension of the lattice. It is possible to write a $n \times m$ matrix A associated to a lattice, in which the rows of the matrix are the coordinates of the vectors of the basis.

In order to expose the most famous lattice problems, on which lattice cryptography is based, we need to introduce the minimal distance $\lambda_1(L)$, that is $\lambda_1(L) = \min_{v \in L \setminus \{0\}} ||v||$. Analogously it is possible to define the successive minimum $\lambda_i(L)$ for $2 \leq i \leq n$ as

$$\lambda_i(L) = \min_r \{v_1, \ldots, v_i \text{ independent in } L : ||v_j|| \leq r \text{ for } 1 \leq j \leq i\}.$$

Moreover we need to consider other algebraic structures: given a polynomial $\phi(x) \in \mathbb{Z}[x]$, usually $\phi(x) = x^n - 1$ or $\phi(x) = x^n + 1$, and a prime $q > 2$, we define $R = \mathbb{Z}[x]/\left(\phi(x)\right)$ and $R_q = R/qR = \mathbb{Z}_q[x]/\left(\phi(x)\right)$.

Some famous examples of hard lattice problems are the following:

- (SVP) Given a basis B of a lattice L, find a vector $v \in L$ such that $||v|| = \lambda_1(L)$.
  (Approx-SVP$_\gamma$) Given a basis B of a lattice L find a vector $v \in L$ such that $||v|| \leq \gamma(n)\lambda_1(L)$, where the constant $\gamma(n)$ depends on the dimension of the lattice n.
  (GapSVP$_\gamma$) Given a basis B of a lattice L and a constant d, decide if $\lambda_1(L) \leq d$ or $\lambda_1(L) > \gamma d$.
- (SIVP$_\gamma$) Given a basis B of a lattice L find linearly independent vectors $v_1, \ldots, v_n \in L$ such that $||v_i|| \leq \gamma\lambda_n(L)$ for $1 \leq i \leq n$.
- (SIS$_\beta$, [2,30]) Given a matrix $A \in M_{n\times m}(\mathbb{Z}_q)$, find a vector $z \in \mathbb{Z}_q^m$ such that $Az \equiv 0 \bmod q$ and $||z|| \leq \beta$.
  (RSIS$_\beta$, [27,37]) Given a vector $a \in R_q^m$, find a vector $z \in R_q^m$ such that $\langle z, b\rangle = 0$ and $||z|| \leq \beta$.
  (MSIS$_\beta$, [26]) Given a matrix $A \in M_{n\times m}(R_q)$, find a vector $z \in R_q^m$ such that $Az \equiv 0 \bmod q$ and $||z|| \leq \beta$.

- For a vector $s \in \mathbb{Z}_q^n$ and a discrete Gaussian distribution χ, with width αq for some $\alpha < 1$, the LWE distribution $A_{s,\chi}$ is sampled by choosing $a \in \mathbb{Z}_q^n$ uniformly at random, e drawn with χ and outputting the pair $(a, b = \langle a, s\rangle + e \bmod q)$.
  (LWE, [38]) Given m pairs $(a_1, b_1), \ldots, (a_m, b_m)$ drawn from $A_{s,\chi}$ for a random $s \in \mathbb{Z}_q^n$, find s.
  For $s \in R_q$ the RLWE distribution $A^R_{s,\chi}$ is sampled by choosing $a \in R_q$ uniformly at random, e drawn with χ and outputting the pair $(a, b = s \cdot a + e \bmod q)$.
  (RLWE, [28]) Given m pairs $(a_1, b_1), \ldots, (a_m, b_m)$ drawn from $A^R_{s,\chi}$ for a random $s \in R_q$, find s.
  For a vector $s \in R_q^n$, the MLWE distribution $A^M_{s,\chi}$ is sampled by choosing $a \in R_q^n$ uniformly at random, e drawn with χ and outputting the pair $(a, b = \langle a, s\rangle + e \bmod q)$.
  (MLWE, [8, 26]) Given m pairs $(a_1, b_1), \ldots, (a_m, b_m)$ drawn from $A^M_{s,\chi}$ for a random $s \in R_q^n$, find s.

Although it seems that the problems of the SIS and LWE families are not related to those on lattices, it can be proved [26,28,31,35,40] that, by using several reductions, solving these problems is as least as difficult as solving instances of GapSVP and SIVP.

See [19, 36, 39] for further details on lattices and lattice cryptography.

1.3. Multivariate polynomial systems theory

Definition 3. Let m, n, q be three positive integers with $m \leq n$ and $\mathbb{F}_q$ a finite field of cardinality q. Let $p_1, \ldots, p_m \in \mathbb{F}_q[x_1, \ldots, x_n]$ be m quadratic polynomials in n variables. The MQ (multivariate quadratic) problem consists of finding a solution $\bar{x} \in {\mathbb{F}_q}^n$ of the system:

$$p_1(x_1, \ldots, x_n) = p_2(x_1, \ldots, x_n) = \ldots = p_m(x_1, \ldots, x_n) = 0. \quad (1)$$

When the system (1) is random, i.e. for all $i = 1, \ldots, m$ the coefficients of p_i are choosen uniformly at random, MQ has been proven to be an NP-hard problem [34]. Since a quantum algorithm for solving MQ problem does not exist, the multivariate protocols are very used in Post-Quantum cryptography.

The field $\mathbb{F}_q$ is not algebraically closed, therefore a solution of (1) definitely belongs to ${\mathbb{F}_q}^n$ if the field equations are added to the system. So, given the polynomial ideal $I = \langle p_1, \ldots, p_m\rangle$, if the solution of

(1) is unique, solving the MQ problem is equivalent to find the only point of the variety $V(I) \cap \mathbb{F}_q{}^n = V(I + \langle x_1^q - x_1, \ldots, x_n^q - x_n \rangle)$ [25].

In terms of functions, solving the MQ problem is equivalent to invert $\mathcal{P} : (x_1, \ldots, x_n) \longmapsto (p_1(x_1, \ldots, x_n), \ldots, p_m(x_1, \ldots, x_n))$, i.e. given $d = \mathcal{P}(z) \in \mathbb{F}_q{}^m$, it is unfeasible to recover $z \in \mathbb{F}_q{}^n$ if it is not known the way in which the polynomials p_i are generated.

In a multivariate public key cryptosystem (MPKC), $\mathcal{P}$ is obtained using a secret set of m quadratic polynomials with random coefficients $\{f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n)\}$ and composing the quadratic map

$$\mathcal{F} : (x_1, \ldots, x_n) \longmapsto (f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n))$$

with two affine maps $\mathcal{S}$ and $\mathcal{T}$. The quadratic map $\mathcal{F}$ is easy to invert [12], but the action of the affine maps makes difficult to invert $\mathcal{P}$, because the polynomials p_i induced by $\mathcal{P}$ are approximately random [5].

A system induced from the previous construction is known as bipolar system [12], and it determines different MPKCs depending on the particular choice of $\mathcal{S}, \mathcal{F}$ and $\mathcal{T}$.

1.4. One-Way functions

A one-way function is any function which can be efficiently computed but whose whose pseudo-inverse is hard to find. More formally:

Definition 4. A function f is said to be one-way if it can be computed in polynomial time on any input x and if any polynomial-time probabilistic algorithm used to solve $f(x) = y$ knowing f and y succeds with negligible probability.

Examples of one-way functions are (cryptographically secure) block ciphers and hash functions. These primitives can therefore be safely used in the design of post-quantum digital signatures, since the only known speed-up for a quantum computer is Grover's search algorithm [18], which however is not capable of determining in polynomial time a pre-image of a one-way function with more than negligible probability.

1.5. Zero-Knowledge proofs

Zero-knowledge proofs (ZKP) are protocols in which a prover can convince a verifier that a statement is true, without disclosing any information apart that the statement is true. The three classic properties that a ZKP needs are

- completeness: honest verifiers will be convinced by honest provers.
- soundness: no malicious prover can prove (with non-negligible probability) a false statement.
- zero-knowledge: no verifier learns anything other than the fact that the statement is true.

2. Signature Schemes

2.1. Rainbow

Rainbow [13] is a generalisation of the Unbalanced Oil and Vinegar (UOV) signature scheme [24], obtained by considering multiple UOV layers. The security of Rainbow is linked to the NP-hard problem of solving a multivariate polynomial system of quadratic equations over the field $\mathbb{F} = \mathbb{F}_{2^s}$. The fundamental parameters of Rainbow are s, three positive integers v_1, o_1 and o_2, and a hash function H whose digest is $(o_1 + o_2) \cdot 2^s$ bits long.

Define two constants $m = o_1 + o_2$ and $n = v_1 + o_1 + o_2 = m + v_1$ and let $V_1 = \{1, \dots, v_1\}$, $V_2 = \{1, \dots, v_1 + o_1\}$, $O_1 = \{v_1 + 1, \dots, v_1 + o_1\}$ and $O_2 = \{v_1 + o_1 + 1, \dots, n\}$ be four sets of integers determined by the parameters, and let $\mathcal{S} : \mathbb{F}^m \longrightarrow \mathbb{F}^m$ and $\mathcal{T} : \mathbb{F}^n \longrightarrow \mathbb{F}^n$ be two invertible affine maps. For each $k \in O_1 \cup O_2$ define the map $f_k : \mathbb{F}^n \to \mathbb{F}$ according to the formula

$$f_k(x_1, \dots, x_n) = \sum_{i,j \in V_l, i \leq j} \alpha_{k,i,j} x_i x_j + \sum_{i \in V_l, j \in O_l} \beta_{k,i,j} x_i x_j + \\ + \sum_{i \in V_l \cup O_l} \gamma_{k,i,j} x_i + \delta_k \ ,$$

where $l \in \{1, 2\}$ is the unique index for which $k \in O_l$ and $\alpha_{k,i,j}$, $\beta_{k,i,j}$, $\gamma_{k,i,j}$, $\delta_k \in \mathbb{F}$ are randomly generated parameters. The m functions in n variables $f_{v_1+1}, \dots, f_n$ are used to define a quadratic map $\mathcal{F} : \mathbb{F}^n \longrightarrow \mathbb{F}^m$, such that

$$\mathcal{F}(x_1, \dots, x_n) = (f_{v_1+1}(x_1, \dots, x_n), \dots, f_n(x_1, \dots, x_n)).$$

Due to the structure of $\mathcal{F}$, given $d \in \mathbb{F}^m$ it is possible to find in a reasonable amount time a value $\bar{z} \in \mathbb{F}^n$ such that $\mathcal{F}(\bar{z}) = d$, employing an algorithm that fixes the first variables and then applies Gaussian elimination. This property is used to efficiently compute a value z such that $\mathcal{P}(z) = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T}(z) = d$. On the other hand, given z and $\mathcal{P}$ it is easy to compute $d = \mathcal{P}(z)$, but from $d \in \mathbb{F}^m$ it is unfeasible to obtain a value $z \in \mathbb{F}^n$ for which $\mathcal{P}(z) = d$ without knowing $\mathcal{F}$ if $\mathcal{S}, \mathcal{T}$, and $\mathcal{F}$ are random.

Given the parameters (s, v_1, o_1, o_2, H) described above, the protocol works as follows.

Key generation

a) Randomly choose $\mathcal{S}$, $\mathcal{T}$ and $\mathcal{F}$ as defined above, choosing the maps' coefficients uniformly at random in $\mathbb{F}$.
b) The private key consists of $(\mathcal{S}, \mathcal{F}, \mathcal{T})$.
c) The public key is the composition $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T}$.

Signing

Given a key-pair $((\mathcal{S}, \mathcal{F}, \mathcal{T}), \mathcal{P})$ and a message digest d, compute the signature performing the following steps:

a) Choose uniformly at random a bit string r with the same length of d.
b) Compute $h = H(d||r)$ interpreted as a vector of $\mathbb{F}^m$.
c) Compute $x = \mathcal{S}^{-1}(h)$.
d) Compute $y = \mathcal{F}^{-1}(x)$.
e) Compute $z = \mathcal{T}^{-1}(y)$.
f) The signature is the pair (z, r).

Verification

To verify a signature (z, r) on a message digest d perform the following steps:

a) Compute $h = H(d||r)$ interpreted as a vector of $\mathbb{F}^m$.
b) Compute $h' = \mathcal{P}(z)$ and check if $h' = h$.

2.2. GeMSS

GeMSS [10] is a multivariate signature scheme, based on a system of polynomial equations over the field $\mathbb{F}_2$. The fundamental parameters of GeMSS are the following: m the number of equations, Δ and v

that determine the number of total variables, and a hash function H whose digest is k bits long.
Fix $n = m + \Delta$ and let $S \in \mathrm{GL}_{n+v}(\mathbb{F}_2)$ and $T \in \mathrm{GL}_n(\mathbb{F}_2)$ be two invertible matrices. Define $F \in \mathbb{F}_{2^n}[X, v_1, \dots, v_v]$, a polynomial of degree D, with the following structure:

$$F(X, v_1, \dots, v_v) = \sum_{\substack{0 \le j < i < n \\ 2^i + 2^j \le D}} A_{i,j} X^{2^i + 2^j} + \sum_{\substack{0 \le i < n \\ 2^i \le D}} \beta_i(v_1, \dots, v_v) X^{2^i} + \\ + \gamma(v_1, \dots, v_v),$$

where $A_{i,j} \in \mathbb{F}_{2^n}$, each $\beta_i : {\mathbb{F}_2}^v \longrightarrow {\mathbb{F}_2}^n$ is linear and $\gamma(v_1, \dots, v_v) : {\mathbb{F}_2}^v \longrightarrow {\mathbb{F}_2}^n$ is quadratic.
Let $(\theta_1, \dots, \theta_n) \in {\mathbb{F}_{2^n}}^n$ be a basis of ${\mathbb{F}_2}^n$ over $\mathbb{F}_2$. Given $E = \sum_{k=1}^n e_k \cdot \theta_k \in \mathbb{F}_{2^n}$, define the following function :

$$\Phi : \mathbb{F}_{2^n} \longrightarrow {\mathbb{F}_2}^n \quad E \longmapsto \Phi(E) = (e_1, \dots, e_n). \tag{2}$$

Starting from F, it is possible to define n multivariate polynomials $f_k \in \mathbb{F}_2[x_1, \dots, x_{n+v}]$, such that $F\big(\sum_{k=1}^n \theta_k x_k, v_1 \dots, v_v\big) = \sum_{k=1}^n \theta_k f_k$. The public key P is derived from $f_1, \dots, f_n$ and it consists of the first m components of

$$(p_1, \dots, p_n) = (f_1((x_1, \dots, x_{n+v}) \cdot S), \dots, f_n((x_1, \dots, x_{n+v}) \cdot S)) \cdot T, \tag{3}$$

which is reduced modulo the field equations, that is $\mathrm{mod}\langle x_1^2 - x_1, \dots, x_n^2 - x_n\rangle$. Due to the structure of F, given $d \in {\mathbb{F}_2}^m$ and $r \in {\mathbb{F}_2}^{n-m}$ randomly chosen, it is possible, with a procedure that fixes the last v variables and then applies Berlekamp's algorithm on the resulting univariate polynomial, to find a root of $F - \Phi^{-1}((d, r) \cdot T^{-1})$ in a reasonable amount time ($O(nD)$). This property is used to efficiently compute a value $z \in {\mathbb{F}_2}^{n+v}$ such that $P(z) = d$. On the other hand, given z and P it is easy to compute $d = P(z)$, but from $d \in {\mathbb{F}_2}^m$ it is unfeasible to obtain a value $z \in {\mathbb{F}_2}^{n+v}$ for which $P(z) = d$ without knowing F, if S,T and F are random.
Finally, it is possible to iterate t times a part of the signature to increase the security level λ, indeed in this way it is possible to apply the hash function H and at the same time to combine the actions of S and T on the variables more than once.

Given the parameters (m, Δ, v, D, H, t) described above, the protocol works as follows.

Key generation

a) Randomly choose S, T and F choosing the coefficients of F uniformly at random in $\mathbb{F}_{2^n}$ and the elements of S and T in $\mathbb{F}_2$.
b) The private key consists of (S, T, F).
c) Compute $p = (p_1, \ldots, p_n)$ as defined in (3).
d) The public key is $P = (p_1, \ldots, p_m)$, the first m components of p.

Signing

Given a key-pair $((S, T, F), P)$ and a message digest h, compute the signature performing the following steps:

a) Set $S_0 = 0 \in {\mathbb{F}_2}^m$.
b) Repeat for $i = 1$ to t the following steps:
   i) Get D_i the first m bits of h and compute $D'_i = D_i \oplus S_{i-1}$.
   ii) Randomly choose $(v_1, \ldots, v_v) \in {\mathbb{F}_2}^v$ and $r \in {\mathbb{F}_2}^{n-m}$.
   iii) Compute $A_i = \phi^{-1}((D'_i, r) \cdot T^{-1})$ as described in (2).
   iv) Compute a root Z of $F - A_i$.
   v) Compute $(S_i, X_i) = (\phi(Z), v_1, \ldots, v_v) \cdot S^{-1} \in {\mathbb{F}_2}^m \times {\mathbb{F}_2}^{n+v-m}$.
   vi) Compute $h = H(h)$.
c) The signature is $z = (S_t, X_t, \ldots, X_1)$.

Verification

To verify a signature z on a message digest h perform the following steps:

a) Repeat for $i = 1$ to t
   i) Get D_i the first m bits of h.
   ii) Compute $h = H(h)$.
b) Repeat for $i = t - 1$ to 0
   i) Compute $S_i = P(S_{i+1}, X_{i+1}) \oplus D_{i+1}$.
c) Check if $S_0 = 0$.

2.3. CRYSTALS-DILITHIUM

CRYSTALS-DILITHIUM [4] is a lattice-based signature built on the hardness of two problems: MLWE and SelfTargetMSIS problem [23], a variation of the MSIS problem. The first problem is defined over a polynomial ring $R_q = \mathbb{Z}_q[x]/(x^{256}+1)$, where q is a prime such that $q \equiv 1 \mod 512$. This condition on q allows to use the NTT (Number Theoretic Transform, a generalization of the discrete Fourier transform over a finite field) representation. Given H a hash function and a vector x, SelfTargetMSIS consists in finding a vector $z' = (z, c)$ with small coefficients such that $H(x||f(z')) = c$ with $\mathtt{w}(c) = 60$ (where $\mathtt{w}$ denotes the Hamming weight), where f is a linear function.

In order to define an ordering relation in $\mathbb{Z}_q$, we will consider the embedding $\eta : \mathbb{Z}_q \to \mathbb{Z}$, where $\eta(z) \equiv z \bmod q$ and $-\frac{q-1}{2} \leq \eta(z) \leq \frac{q-1}{2}$. For any z_1, $z_2 \in \mathbb{Z}_q$ we say that $z_1 \leq z_2$ if and only if $\eta(z_1) \leq \eta(z_2)$.

Let $w = w_0 + w_1x + \ldots + w_{255}x^{255}$ be a polynomial in R_q, the norm $\|w\|_\infty := \max_i(|w_i|)$ is used to check some conditions related to the security and correctness, for this reason it is introduced a parameter $\beta \in \mathbb{Z}$ as a bound for the norm of some quantities. Let $A \in M_{k,l}(R_q)$ be a matrix and set $\bar{w} = Ay$, where $y \in R_q^l$ is a vector such that $\|y\|_\infty \leq \gamma_1$ (with $\gamma_1 \in \mathbb{Z}$ another parameter), we distinguish between the high-order and low-order parts of $\bar{w}$ as follows: for each component w' of $\bar{w}$

$$w' = w'_1 \cdot 2\gamma_2 + w'_0, \tag{4}$$

where $\|w'_0\|_\infty \leq \gamma_2$, where $\gamma_2 \in \mathbb{Z}$ is another parameter. We call w'_1 the high-order part, while w'_0 is the low-order part of w'. We denote with HB$(\bar{w})$ (HighBits) the vector comprising all w'_1s, thus is the high-order part of $\bar{w}$, and with LB$(\bar{w})$ (LowBits) the low-order part of $\bar{w}$.

For storage efficiency, instead of generating and storing the entire matrix A, the protocol makes use of a secure PRNG and the NTT: using the NTT, it is possible to identify $a \in R_q$ and $\bar{a} \in \mathbb{Z}_q^{256}$, where $\bar{a}$ is the NTT representation of a, while if $A \in R_q^{k\times l}$ we denote with NTT(A) the matrix where each coefficient of A is identified with an element of ${\mathbb{Z}_q}^{256}$. To obtain the matrix A, every element $\bar{a}_{i,j}$ of NTT(A) is generated from a 256 bit random seed ρ.

The parameters of CRYSTALS-DILITHIUM are q, d, k, l, η, γ_1, γ_2, Ω, H, G, where γ_1, γ_2, k, l and q are defined as above, $d \in \mathbb{Z}_q$, η and Ω are other bounds and G, H are hash functions.

Given the parameters $(q, k, l, \eta, G, H, d, \gamma_1, \gamma_2, \beta, \Omega)$ described above, the protocol works as follows.

Key generation

a) Choose uniformly at random two bit strings ρ and θ of length 256.
b) Choose uniformly at random $(s_1, s_2) \in {R_q}^l \times {R_q}^k$ with $|s_i| \leq \eta$.
c) Compute $A \in R_q^{k\times l}$ from ρ using NTT representation.
d) Compute $t = As_1 + s_2$.
e) Compute $t_0 = t \ \mathtt{mod}\ 2^d$ and $t_1 = \frac{t-t_0}{2^d}$.
f) The public key is $P = (\rho, t_1)$.
g) The private key is $S = (\rho, \theta, G(\rho||t_1), s_1, s_2, t_0)$.

Signing

Given a key-pair (S, P) and a message M compute the signature performing the following steps:

a) Compute A from ρ as described above.
b) Compute $\mu = G(G(\rho||t_1)||M)$ and $\rho' = G(\theta||\mu)$.
c) Compute uniformly at random $y \in R_q^l$ with $\|y\|_\infty < \gamma_1$, starting from seed ρ' using NTT representation.
d) Compute $w = Ay$ and $w_1 = \mathrm{HB}(w)$.
e) Compute $c = H(\mu||w_1)$ and $z = y + cs_1$.
f) Compute $r_1 = \mathrm{HB}(w - cs_2)$ and $r_0 = \mathrm{LB}(w - cs_2)$.
g) Check if all the following conditions are satisfied else repeat from step 3:

    i) $\|z\|_\infty < \gamma_1 - \beta$.
    ii) $\|r_0\|_\infty < \gamma_2 - \beta$.
    iii) $r_1 = w_1$.

h) Compute $h = (h_1, \ldots, h_k) = r_1 \oplus \mathrm{HB}(w - cs_2 + ct_0)$.
i) Compute $\Omega' = \mathtt{w}(h)$ and check if $\Omega' \leq \Omega$ else repeat from step 3.
j) The signature is (z, h, c).

Verification

To verify a signature (z, h, c) on a message M perform the following steps:

a) Compute A and μ as described in the signing process.

b) Compute $w'_1 = \mathrm{HB}(w - cs_2)$ knowing $\mathrm{HB}(Az - ct_1 \cdot 2^d) = \mathrm{HB}(w - cs_2 + ct_0)$ and h that allows to remove the error generated by ct_0.

c) Check if all the following conditions are satisfied:

   i) $\|z\|_\infty < \gamma_1 - \beta$.

   ii) $c = H(\mu||w'_1)$.

   iii) Compute $\Omega' = \mathtt{w}(\mathtt{h})$ and check if $\Omega' \leq \Omega$.

Given the parameter $d \in \mathbb{Z}_q$ and computed $z = y + cs_1$ with $s_1 \in R_q^l$, it is possible to define $t_1, t_0 \in \mathbb{Z}_q$ such that $t = t_1 \cdot 2^d + t_0$ (t_1 is the high order part of t) and compute $\mathrm{HB}(\bar{w} - cs_2 + ct_0)$. Indeed:

$$Az - ct_1 \cdot 2^d = Ay + cAs_1 - c(t - t_0) = Ay - cs_2 + ct_0 = \bar{w} - cs_2 + ct_0.$$

Starting from $r_1 = \mathrm{HB}(\bar{w} - cs_2 + ct_0)$, it is easy to obtain $\mathrm{HB}(\bar{w} - cs_2)$ knowing $h = r_1 \oplus \mathrm{HB}(\bar{w} - cs_2)$, indeed it is sufficient to check which bits of h have value 1 to find the error bits in r_1 and changing their value. The arithmetic modulus $\frac{q-1}{2\gamma_2}$ is required to modify r_1 depending on the sign of $\mathrm{LB}(\bar{w} - cs_2 + ct_0)$. Besides, the parameter Ω is the maximum Hamming weight that h can assume and thanks to the condition $\|z\|_\infty < \gamma_1 - \beta$, it is possible to make the correction of error bits successfully, in a safe way. On the other hand, it is infeasible to recover z without knowing y (so $\bar{w}$ cannot be computed) and s_1.

2.4. FALCON

FALCON [16] is a particular lattice-based signature, which is based on solving the SIS problem over the NTRU lattices. Given $n = 2^k$, $q \in \mathbb{N}^*$ and defined R using $\phi(x) = x^n + 1$, the problem consists in determining $f, g, G, F \in R$ such that f is invertible modulus q (this condition is equivalent to require that $\mathrm{NTT}(f)$ does not contain 0 as a coefficient) and such that the following equation (NTRU equation) is satisfied:

$$fG - gF = q \mod \phi. \tag{5}$$

If $h := g \cdot f^{-1} \mod q$, it is possible to verify that the matrices $P = \begin{bmatrix} 1 & h \\ 0 & q \end{bmatrix}$ and $Q = \begin{bmatrix} f & g \\ F & G \end{bmatrix}$ generate the same lattice:

$$\Lambda(P) = \{zP \mid z \in R_q\} = \{zQ \mid z \in R_q\} = \Lambda(Q),$$

but, if f and g are sufficiently small, then h should seem random, so, given h, the hardness of this problem consists of finding f and g. Each coefficient of the polynomials $f = \sum_{i=0}^{n-1} f_i x^i$ and $g = \sum_{i=0}^{n-1} g_i x^i$ is generated from a distribution close to a Gaussian of center 0 and standard deviation $\sigma \in [\sigma_{\min}, \sigma_{\max}]$ (where $\sigma, \sigma_{\min}, \sigma_{\max}$ are parameters).

The following general property is fundamental to solve the NTRU equation 5, in particular if $f = \sum_{i=0}^{n-1} a_i x^i \in \mathbb{Q}[x]$, f can be decomposed in a unique way as:

$$f(x) = f_0(x^2) + x f_1(x^2), \tag{6}$$

where $f_0 = \sum_{i=0}^{n/2-1} a_{2i} x^i$ and $f_1 = \sum_{i=0}^{n/2-1} a_{2i+1} x^i$.

Given f and g, it is easy to obtain (F, G) solution of (5), indeed there is a recursive procedure that uses the previous property and allows to solve a NTRU equation in the ring $\mathbb{Z} = \mathbb{Z}[x]/(x+1)$ and then transforms this solution $(F, G) \in \mathbb{Z} \times \mathbb{Z}$ into two polynomials of $\mathbb{Z}[x]/(\phi)$. Thanks to the FFT, it is possible to define the matrix $\bar{B} = \begin{bmatrix} \mathrm{FFT}(g) & \mathrm{FFT}(-f) \\ \mathrm{FFT}(G) & \mathrm{FFT}(-F) \end{bmatrix}$. Moreover we also need to consider the LDL decomposition of $\mathcal{G} = \bar{B} \cdot \bar{B}^T = LDL^T$, where $L = \begin{bmatrix} 1 & 0 \\ \bar{L} & 1 \end{bmatrix}$ and $D = \begin{bmatrix} D_{11} & 0 \\ 0 & D_{22} \end{bmatrix}$.

Starting from $\mathcal{G} \in M_{2,2}\left(\mathbb{Q}[x]/(\phi)\right)$, it is possible to construct the so-called FALCON tree T: the root of T is $\bar{L}$ and its two child-nodes $G_0, G_1 \in M_{2,2}\left(\mathbb{Q}[x]/\left(x^{n/2}+1\right)\right)$ are obtained considering the decomposition of D_{11} and D_{22} as described in (6). Iterating this procedure on G_0 and G_1, it is possible to obtain the whole tree T, where each leaf $l \in \mathbb{Q}$ is normalized, i.e $l' = \frac{\sigma}{l}$.

FALCON uses a particular hash function H, which transforms a string modulus q in a polynomial $c \in \mathbb{Z}_q[x]/(\phi)$.

In addition to the standard deviations described above, the parameters of FALCON are k, q and two other constants $\beta \in \mathbb{Q}_+$ and $b_l \in \mathbb{N}^*$ that will be described later.

Given a solution t of $\bar{B}t = c$, there exists a recursive procedure (Fast Fourier sampling), which applies a randomized rounding on

the coefficients of $t \in \mathbb{Q}[x]/(\phi)$ to obtain a polynomial $z \in R$, using the information stored in T.
Let $a, b \in \mathbb{Q}[x]/(\phi)$, it is possible define the following inner product and its associated norm:

$$< a, b >= \frac{1}{n} \sum_{i \in \mathbb{C} \,:\, \phi(i)=0} a(i) \cdot \overline{b(i)}.$$

Let $\beta \in \mathbb{Q}_+$, then it is possible to compute $s = (t - z)\bar{B}$ with $||s||^2 \leq \lfloor \beta^2 \rfloor$ and using the inverse of FFT it is easy to compute $s_1, s_2 \in R$, which satisfy:

$$s_1 + s_2 h = c \bmod q. \tag{7}$$

On the other hand, given (s_1, s_2) it is unfeasible to recover s without knowing $\bar{B}$ and T.
Finally, FALCON uses a compression algorithm, which transforms s_2 in a byte string $(8 \cdot b_l - 328)$ long.

Given the parameters $(k, q, \sigma_{\min}, \sigma_{\max}, \sigma, \beta, b_l)$ described above, the protocol works as follows.

Key generation

a) Compute $f = \sum_{i=0}^{n-1} f_i x_i$ and $g = \sum_{i=0}^{n-1} g_i x_i$, generating f_i and g_i from Gaussian distribution $D_{0,\sigma}$.
b) Check that f is invertible modulus q, else restart from step 1.
c) Find (F, G) solution of the NTRU equation (5).
d) Compute $\bar{B}$ as described above.
e) Compute $G = \bar{B} \cdot \bar{B}^T$, obtain the FALCON tree T using LDL decomposition and normalize its leaves.
f) Compute $h = gf^{-1} \bmod q$.
g) The private key is $(\bar{B}, T)$.
h) The public key is h.

Signing

Given a key-pair $(h, (\bar{B}, T))$ and a message m, compute the signature performing the following steps:

a) Choose uniformly at random a bit string r 320 long.
b) Compute $c = H(r||m, q, n)$ and solve $\bar{B}t = c$.

c) Compute z randomized rounding of t as described above.
d) Compute $s = (t - z)\bar{B}$.
e) Check that $||s||^2 \leq \lfloor \beta^2 \rfloor$ else repeat from step 3.
f) Compute (s_1, s_2) satisfying (7).
g) By compressing s_2, compute a string s' of $(8 \cdot b_l - 328)$ bytes.
h) The signature is (r, s').

Verification

Given a public key h, to verify a signature (r, s') on a message digest c perform the following steps:

a) By decompressing s', compute s_2.
b) Compute $s_1 = c - s_2 h \bmod q$.
c) Check if $||(s_1, s_2)||^2 \leq \lfloor \beta^2 \rfloor$.

2.5. SPHINCS$^+$

SPHINCS$^+$ [3] is based on hash functions and it is nothing more than an opportune union of three signature schemes: WOTS$^+$ [20], XMSS [9] and FORS [7].
SPHINCS$^+$ works with two main tree structures: a Hypertree and a FORS tree. The Hypertree consists of d Merkle trees of height h'. On each of these trees is applied an XMSS signature scheme. XMSS, in turn, consists of a one-time signature WOTS$^+$ applied on the root of the previous layer plus the authentication path of the randomly chosen leaf.
On the other hand, a FORS tree is made up of k parallel trees of height a and, contrary to Hypertrees, this kind of trees is used only on signature generation and verification, but not for key generation. SPHINCS$^+$ uses the FORS scheme to generate a hash value that relates the message to k FORS roots. After that, a Hypertree signature is applied to the hash returned by the FORS signature to generate a SPHINCS$^+$ signature.

The security of this scheme derives from the security of the hash function involved. In particular SPHINCS$^+$ uses the so called tweakable hash functions, which allow us to approach the details of how exactly the nodes are computed.
The choice of the hash function strongly influences the security of the signature, in fact the length n of every hash value of this protocol is fundamental to determinate the security level, the authors have chosen SHAKE256 as the hash function family.

The parameters k and a determine the performance and security of FORS, so it is necessary to balance the value of these two parameters to avoid getting too large or too slow signatures. Instead, the height of the Hypertree $h'd$ determines the number of XMSS instances, so this value has a direct impact on security: a taller Hypertree gives more security. Remark that the number of layers d is a pure performance trade-off parameter and does not influence security. Finally, the Winternitz parameter w is a trade-off parameter (greater w means shorter signatures but slower signing), which determines the number and length of the hash chains per WOTS$^+$ instance. The privacy of SPHINCS$^+$ is guaranteed by the pseudorandom generation of WOTS$^+$ and FORS secret keys (this operation randomizes the choice of FORS and WOTS$^+$ leaves used to sign).

Given the parameters (n, h', d, k, a, w) described above, the protocol works as follows.

Key generation

The description of key generation assumes the existence of a function secRand which on input n returns n bytes of cryptographically strong randomness.

a) Compute SK.seed=secRand(n), which is used to generate all the WOTS$^+$ and FORS private key elements.
b) Compute SK.prf=secRand(n), which is used to generate a randomization value for the randomized message hash.
c) Compute PK.seed=secRand(n), which is the public seed.
d) Compute PK.root, which is the hypertree root, i.e. the XMSS root of the tree on the top level.
e) The private key is: SK=(SK.seed, SK.prf, PK.seed, PK.root).
f) The public key is: PK=(PK.seed, PK.root).

Signing

Given the private key SK and a message M, compute the signature performing the following steps:

a) Compute R, an n-bytes string pseudorandomly generated starting from SK.prf and M.
b) Compute the digest of M.
c) Compute SIGFORS, which is a FORS signature applied to the first ka bits of the digest.

d) Starting from SIGFORS, derive PKFORS i.e. the public key associated to the FORS signature.
e) Compute HTSIG, which is an hypertree signature applied to PKFORS.
f) The SPHINCS$^+$ signature is: SIG=(R, SIGFORS, HTSIG).

Verification

To verify a signature SIG on a message M perform the following steps:

a) Get R, which corresponds to the first n bytes of SIG.
b) Get SIGFORS, the following $k(a+1) \cdot n$ bytes of SIG.
c) Compute the digest of M.
d) Starting from SIGFORS and the first ka bits of the digest, derive PKFORS.
e) Starting from PKFORS, check the hypertree verification.

2.6. Picnic

Picnic [11] is a signature scheme whose security is based on the one-wayness of a block cipher and the pseudo-random properties of an extensible hash function.

In particular the construction relies upon the fact that a digital signature is essentially a non-interactive zero knowledge proof of knowledge of the preimage of a one-way function output, where the challenge inside the proof is tied to the message that is being signed. In other words, the signer creates a transcript that demonstrates the knowledge of the private key whose image through the one-way function is the public key, without revealing any information about the private key itself. Moreover this transcript is indissolubly bound to the message.

Starting from this general idea, Picnic instantiates a signature scheme using classical general-purpose primitives: a block cipher, a secure multi-party computation protocol (MPC), and an extensible cryptographic hash function (also known as extensible output function or XOF). The zero-knowledge proof (ZKP) is derived from the hash and the MPC protocol, exploiting the security properties of the latter. The prover computes the one-way function using its multi-party decomposition, controlling every party. The security of the MPC protocol allows the disclosure of the complete view of some (in this case all but one) parties without revealing anything about the secret input, so a ZKP may be constructed committing to every view and randomly selecting which ones to reveal (the challenge). The commitment (built from the hash) binds the prover to the views (i.e. they cannot be changed after the commitment) without revealing them yet (the commitment is hiding). Soundness can be achieved repeating this process for a few iterations, so that the verifier can be convinced that the prover could not have successfully produced the views without actually knowing the MPC input, except with negligible probability.

The protocol just described is interactive, but there are fairly simple techniques that allow to transform it into a non-interactive one, i.e. a transcript produced by the prover that by itself can convince a verifier. These techniques use a deterministic pseudorandom generator (the hash) to derive the challenges from the public values, i.e. the public key, the commitments and the message. Assuming the (quantum) random oracle model [6, 43] (i.e. the hash is modeled through an oracle that outputs random values on new inputs, but does not change answer when a query is repeated), we maintain soundness even without interaction, and the message is tightly fastened to the transcript, so that it is infeasible to adapt this signature for another message without knowing the private key.

The MPC protocols are much more sensitive to the number of AND operations on two secret bits than to XOR operations, since the masking of AND gates requires extra information to keep consistency. This, in turn, causes the MPC views (and thus the signature) to grow in size, therefore Picnic selected as block cipher LowMC [1], an algorithm designed to minimize such operations for a given security level. LowMC employs a classic substitution-permutation structure with n-bit blocks (where n essentially defines the security level of the whole signature) and r rounds in which s parallel 3-bit S-boxes are applied (note that they do not necessarily cover the entire block), followed by a linear permutation (defined by a different matrix for each round), and a round-key addition (the round-keys are derived multiplying the key by $r + 1$ different matrices: one for the initial key-whitening and again one per round).

Picnic's NIST submission defines various parameters sets that, besides optimizing LowMC parameters for the three security levels, employ different MPC protocols and techniques to obtain a non-interactive zero-knowledge proof (NIZKP). More specifically, `picnic-LX-FS` (where $X \in \{1, 3, 5\}$ is the security level defined by NIST) uses the proof sistem ZKB++ (an optimized version of ZKBoo [17], a ZKP for boolean circuits based on an MPC called "circuit decomposition") that simulates T parallel MPC executions between 3 parties, and uses the Fiat-Shamir transform [15] to obtain a NIZKP. The `picnic-LX-full` variant changes the LowMC parameters: uses a full S-box layer that allows to reduce the number of rounds. The parameter sets `picnic-LX-UR` use again ZKB++ but with the Unruh transform [42–44], which expands the signature size but is provable secure in the stronger quantum random oracle model (unlike the FS transform in general). Finally the sets `picnic3-LX` bring along various optimizations: like `picnic-LX-full` they use a full S-box layer and the Fiat-Shamir transform, but they use a different ZKP and employ various optimizations to reduce signature size. The ZKP used in `picnic3-LX` is the KKW protocol [21], which simulates T parallel MPC executions between N parties ($N = 16$ in the chosen parameters sets). Each execution is divided into an offline preprocessing phase and an online phase where the shares are broadcast and the output reconstructed. In KKW the challenger chooses u executions for which the online phase will be revealed for all but one party, whereas for the other executions only the preprocessing phases will be revealed (for all parties).

Note that the MPC executions assume that each party consumes some random bits read from an input tape. These tapes are deterministically generated from seeds through the XOF, and in turn those seeds are generated from a master seed, which is generated alongside a salt (used as extra input in every other derivation to prevent multi-target attacks such as in [14]) from the secret key, the message, the public key, and the length parameter S (and optionally an extra random input to randomize signatures), always through the XOF. The `picnic3-LX` parameters sets employ a tree structure to derive the seeds in order to reduce the amount of information needed to be included in the signature to reveal the MPC executions. Moreover they use Merkle trees to compute the commitments, so the signatures can be compressed further.

All parameter sets use as XOF an instance of the SHAKE family [33] (specifically SHAKE128 for security level L1 and SHAKE256 for L3 and L5) employing domain separation techniques to differentiate the uses as different random oracles.

Given the parameters (S, n, s, r, T, u) described above, the protocol works as follows.

Key generation

a) Choose a random n-bit string p, and a random n-bit string k.

b) Using LowMC with parameters (n, s, r), compute the encryption of p with k, denoted $C = E(k, p)$.
c) The private key is k.
d) The public key is (C, p).

Signing

Given a key-pair $((C, p), k)$ and a message M, compute the signature performing the following steps:

a) Derive the master seed and the salt from $k, M, (C, p), S$ (and possibly a random input of size $2S$), then derive the individual seeds.
b) Simulate T executions of the MPC protocols, producing for each party their view and output, starting from their seed.
c) Compute the commitments to every seed and corresponding view.
d) Compute the NIZKP challenge e from the MPC outputs, the commitments, the salt, the public key and the message.
e) Compute the NIZKP response by selecting for each MPC execution the appropriate outputs and decommitments to reveal, according to e.
f) Assemble the signature σ by including: e, the salt, the NIZKP response, and the commitments not derivable from the response.

Verification

Given a public key (C, p), to verify a signature σ on a message M perform the following steps:

a) Deserialize σ extracting the NIZKP challenge e, the salt, the NIZKP response, and the commitments.
b) Parse the NIZKP response to obtain, for each of the T MPC executions, the outputs and the decommitments prescribed by e.
c) Use the seeds included in the decommitments to derive (with the salt) the tapes of the revealed parties, then use these values and the rest of the response to simulate the MPC executions that compute the LowMC encryption of p with output C, computing the views for which the commitments are not included in the signature.

d) Complete the commitments deriving the missing values from the results of the previous step, then derive the challenge e' as in signing.

e) The signature is valid if every parsing/deserialization succeeds, the MPC computations are correct, and $e' = e$.

3. Comparison

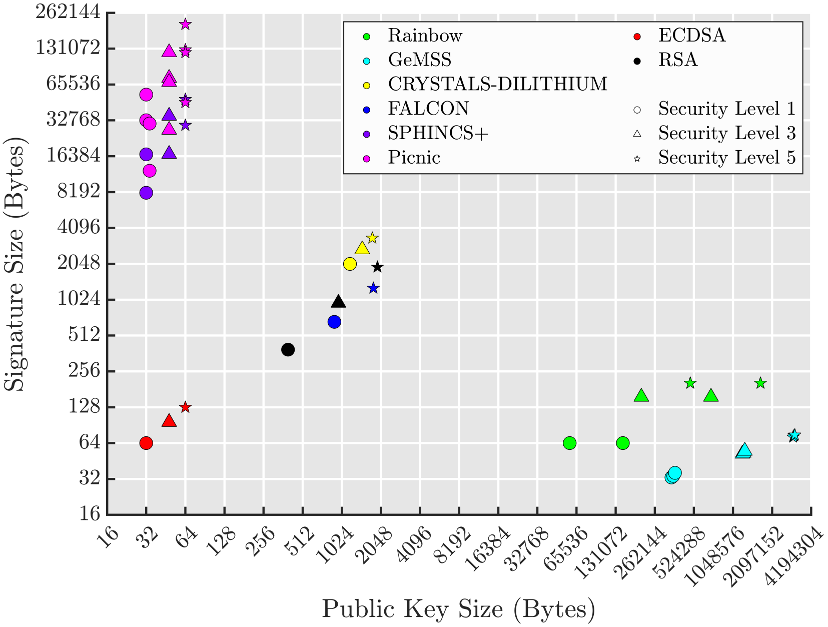


Figura 1. Signature-Public Key Size Comparison

In Figure 1 we summarize the dimensions in bytes of the public keys and the corresponding dimensions of the signatures of all the schemes presented in this survey, as well as those of the two classical schemes ECDSA and RSA. It is interesting to notice that the multivariate schemes have small signatures, but the size of their public keys is the largest among all the schemes. On the other hand, SPHINCS$^+$ and Picnic have small public keys, but large signatures, while the algorithms based on lattices have intermediate values in terms of both public keys and signatures. Finally it is worth to point out that, among all the schemes depicted, the best compromise in terms of dimension is still obtained by the non-quantum scheme of ECDSA.

CrypTOrino 2021
ISBN 000-00-000-0000-0
DOI 10.4399/000000000000021
pag. 93–96 (xx 9999)

Bibliografia

[1] Albrecht, M.R., Rechberger, C., Schneider, T., Tiessen, T. and Zohner, M., Ciphers for MPC and FHE, In Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 430-454), Springer, Berlin, Heidelberg, (2015).

[2] Ajtai, M., Generating hard instances of lattice problems, In Proceedings of the twenty-eighth annual ACM symposium on Theory of computing (pp. 99-108), (1996).

[3] Aumasson, J.P., Bernstein, D.J., Beullens, W., Dobraunig, C., Eichlseder, M., Fluhrer, S., Gazdag, S.L., Hülsing, A., Kampanakis, P., Kölbl, S., Lange, T., Lauridsen, M.M., Mendel, F., Niederhagen, R., Rechberger, C., Rijneveld, J., Schwabe, P., Westerbaan, B., SPHINCS$^+$. Submission to the NIST post-quantum project, v.3, https://sphincs.org/data/sphincs+-round3-specification.pdf, (2020).

[4] Bai, S., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D., CRYSTALS-Dilithium Algorithm Specifications and Supporting Documentation. Round-3 submission to the NIST PQC project, https://pq-crystals.org/dilithium/data/dilithium-specification-round3-20210208.pdf (2021).

[5] Bardet, M., Étude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie These de doctorat de l'Universite Paris 6, Paris(2004)

[6] Bellare M, Rogaway P., Random oracles are practical: A paradigm for designing efficient protocols. Proceedings of the 1st ACM Conference on Computer and Communications Security (pp. 62-73), (1993).

[7] Bernstein, D. J., Hopwood, D., Hülsing, A., Lange, T., Niederhagen, R., Papachristodoulou, L., Schneider, M., Schwabe, P., Wilcox-O'Hearn, Z., SPHINCS: practical stateless hash-based signatures, In Annual international conference on the theory and applications of cryptographic techniques (pp. 368-397), Springer, Berlin, Heidelberg, (2015).

[8] Brakerski, Z., Gentry, C., Vaikuntanathan, V., Fully homomorphic encryption without bootstrapping, ACM Transactions on Computation Theory (TOCT), 6(3), 1-36, (2014).

[9] Buchmann, J., Dahmen, E., Hülsing, A., XMSS-a practical forward secure signature scheme based on minimal security assumptions, In

International Workshop on Post-Quantum Cryptography (pp. 117-129), Springer, Berlin, Heidelberg, (2011).

[10] Casanova, A., Faugere, J.C., Macario-Rat, G., Patarin, J., Perret, L., Ryckeghem, J., GeMSS: a great multivariate short signature, Submission to the NIST's post-quantum cryptography standardization process, https://www-polsys.lip6.fr/Links/NIST/GeMSS_specification_round2.pdf, (2017).

[11] Chase, M., Derler, D., Goldfeder, S., Katz, J., Kolesnikov, V., Orlandi, C., Remacher, S., Rechberger, C., Slamanig, D., Wang, X., Zaverucha, G, The picnic signature scheme, Submission to NIST Post-Quantum Cryptography project, https://github.com/microsoft/Picnic/blob/master/spec/design-v2.2.pdf, (2020).

[12] Ding, J., Gower, J., Schimdt, D., Multivariate Public Key Cryptosystems, Cincinnati: Springer (2006).

[13] Ding, J., Chen, M.S., Kannwischer, M., Patarin, J., Petzoldt, A., Schmidt, D., Yang, B.Y., Rainbow - Algorithm Specification and Documentation, Submission to the NIST's post-quantum cryptography standardization process, https://drive.google.com/file/d/1tcGC38SSkF_csxpzJpkM3qzfsWJq1ywL/view?usp=sharing (2020).

[14] Dinur, I., Kales, D., Promitzer, A., Ramacher, S. and Rechberger, C., Linear equivalence of block ciphers with partial non-linear layers: application to LowMC, In Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 343-372), Springer, Cham (2019).

[15] Fiat, A. and Shamir, A., How to prove yourself: Practical solutions to identification and signature problems, In Conference on the theory and application of cryptographic techniques (pp. 186-194), Springer, Berlin, Heidelberg (1986).

[16] Fouque, P. A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Prest, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z. FALCON: Fast-Fourier lattice-based compact signatures over NTRU. Submission to the NIST's post-quantum cryptography standardization process, https://falcon-sign.info/falcon.pdf (2020).

[17] Giacomelli, I., Madsen, J. and Orlandi, C., Zkboo: Faster zero-knowledge for boolean circuits, In 25th usenix security symposium (usenix security 16), pp. 1069-1083, (2016).

[18] Grover, L. K., A fast quantum mechanical algorithm for database search, Proceedings, in Proceedings of the 28th Annual ACM Symposium on the Theory of Computing, p. 212, (1996).

[19] Hoffstein, J., Pipher, J., Silverman, J. H, An introduction to mathematical cryptography (Vol. 1), New York: Springer (2008).

[20] Hülsing, A., W-OTS$^+$–shorter signatures for hash-based signature schemes, in International Conference on Cryptology in Africa (pp. 173-188), Springer, Berlin, Heidelberg, (2013).

[21] Katz, J., Kolesnikov, V. and Wang, X., Improved non-interactive zero knowledge with applications to post-quantum signatures, In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 525-537 (2018).

[22] Kerry, C., Gallagher, FIPS PUB 186-4 Federal information processing standards publication digital signature standard (DSS), National Institute of Standard and Technology, (2013).

[23] Kiltz, E., Lyubashevsky, V., Schaffner, C., A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model, In Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 552-586), Springer, Cham (2018).

[24] Kipnis, A., Patarin, J., Goubin, L., Unbalanced oil and vinegar signature schemes, In International Conference on the Theory and Applications of Cryptographic Techniques (pp. 206-222), Springer, Berlin, Heidelberg (1999).

[25] Kreuzer, M., Robbiano, L., Computational commutative algebra 1. Springer-Verlag, Berlin (2000).

[26] Langlois, A., Stehlé, D., Worst-case to average-case reductions for module lattices, Designs, Codes and Cryptography, 75(3), 565-599 (2015).

[27] Lyubashevsky, V., Micciancio, D., Generalized compact knapsacks are collision resistant, In International Colloquium on Automata, Languages, and Programming, pp. 144-155, Springer, Berlin, Heidelberg, (2006).

[28] Lyubashevsky, V., Peikert, C., Regev, O., On ideal lattices and learning with errors over rings, In Annual International Conference on the Theory and Applications of Cryptographic Techniques pp. 1-23, Springer, Berlin, Heidelberg, (2010).

[29] Menezes, A., Van Oorschot, P., Vanstone, S., Handbook of applied cryptography, CRC press, (2018).

[30] Micciancio, D., Regev, O., Worst-case to average-case reductions based on Gaussian measures, SIAM Journal on Computing, 37(1), 267-302, (2007).

[31] Micciancio, D., Regev, O., Lattice-based cryptography, In Post-quantum cryptography, pp. 147-191, Springer, Berlin, Heidelberg, (2009).

[32] NIST, Post-Quantum Cryptography Standardization, https : / / csrc . nist . gov / Projects / post-quantum-cryptography /

post-quantum-cryptography-standardization, Accessed: 2021-04-01.

[33] NIST, SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions, National Institute of Standards and Technology (NIST), FIPS PUB 202, U.S. Department of Commerce, (2015).

[34] Patarin J., Goubin L., Trapdoor One-Way Permutations and Multivariate Polynomials, Proceedings of the First International Conference on Information and Communication Security, LNCS 1334, 356-368 (1997).

[35] Peikert, C., Public-key cryptosystems from the worst-case shortest vector problem, In Proceedings of the forty-first annual ACM symposium on Theory of computing, pp. 333-342, (2009).

[36] Peikert, C., A decade of lattice cryptography, Foundations and Trends in Theoretical Computer Science, 10(4), pp. 283-424, (2016).

[37] Peikert, C., Rosen, A., Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices, In Theory of Cryptography Conference (pp. 145-166), Springer, Berlin, Heidelberg, (2006).

[38] Regev, O., On lattices, learning with errors, random linear codes, and cryptography, Journal of the ACM (JACM), 56(6), pp. 1-40, (2009).

[39] Regev, O., The learning with errors problem, Invited survey in CCC, 7(30), 11, (2010).

[40] Regev, O., On lattices, learning with errors, random linear codes, and cryptography, Journal of the ACM (JACM), 56(6), pp. 1-40, (2009).

[41] Stinson, D. R., Paterson, M., Cryptography: theory and practice, CRC press, (2018).

[42] Unruh, D., Quantum proofs of knowledge, In Annual international conference on the theory and applications of cryptographic techniques, pp. 135-152, Springer, Berlin, Heidelberg, (2012).

[43] Unruh, D., Non-interactive zero-knowledge proofs in the quantum random oracle model, In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 755-784, Springer, Berlin, Heidelberg (2015).

[44] Unruh, D., Computationally binding quantum commitments, In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 497-527, Springer, Berlin, Heidelberg, (2016).

Università di Trento
giordano.santilli@unitn.it

CrypTOrino 2021
ISBN 000-00-000-0000-0
DOI 10.4399/000000000000022
pag. 97–?? (xx 9999)

One Time Pad and the Short Key Dream

Umberto Cerruti

Introduction

This is a survey on the One Time Pad (OTP) and its derivatives, from its origins to modern times. OTP, if used correctly, is (the only) cryptographic code that no computing power, present or future, can break. Naturally, the discussion shifts to the creation of long random sequences, starting from short ones, which can be easily shared. We could call it the Short Key Dream. Many problems inevitably arise, which affect many fields of computer science, mathematics and knowledge in general. This work presents a vast bibliography that includes fundamental classical works and current papers on randomness, pseudorandom number generators, compressibility, unpredictability and more.

1. The beginning of modern cryptography

There are several books on the history of cryptography, see [52], [60], [10], [8], [9], [29], [57], [70]. These texts also contain the basics of cryptography. You can find both the history of cryptography and its modern addresses in William Easttom's beautiful book [34].

We can trace the beginnings of modern cryptography back to 1467, the year of the publication of the book De Cifris, by Leon Battista Alberti. In this text, an encryption tool is described in detail. There are two concentric discs, one of which is movable and can rotate inside the other. The symbols of the alphabet are imprinted on the discs, so that a rotation of a certain angle corresponds to a permutation of the alphabet itself. The method therefore consists in replacing one symbol with another, in a reversible way, as in the ancient so called monoalphabetic substitution codes. However, there is a fundamental difference: the permutation may be changed for each letter of the text. For this reason these are called polyalphabetic substitution codes, or Vigenère codes. Blaise de Vigenère, in his book Traicté des Chiffres, written in 1586, divulged and improved the ideas of Alberti and his successors Giovan Battista Bellaso and Giovanni Battista Porta. Vigenère's fundamental contribution can be seen in the explicit use of the key (see [8], Chapter 3).

We describe the polyalphabetic substitution method with modern terminology.

We call alphabet a set $\mathcal{A}$ of q symbols. $\mathcal{S}$ is the set of all permutations of $\mathcal{A}$. A string of permutations $s = \sigma_1 \sigma_2 \ldots \sigma_k$, where $\sigma_i \in \mathcal{S}$, is a key. We call k the length of the key. A message M is a string of symbols from $\mathcal{A}$, $M = x_1 x_2 \ldots x_t$.

M is encoded by the function E_s in this way

$$E_s(M) = \sigma_1(x_1)\,\sigma_2(x_2)\ldots\sigma_k(x_k)\,\sigma_1(x_{k+1})\,\sigma_2(x_{k+2})\ldots \tag{8}$$

Vigenère's idea is very effective. The statistic of the message is completely destroyed. The statistic of the message is the frequency distribution of the q symbols that appear in the message. An investigation based on the relative frequencies of letters is useless. However in 1863 Friedrich Kasiski realized that, if the keyword length k is known, the problem of breaking a polyalphabetic substitution code can be reduced to that of deciphering k monoalphabetic codes.

In fact the symbols $x_1, x_{k+1}, x_{2k+1}, \ldots, x_{hk+1}$ will be encrypted by the permutation σ_1, the symbols $x_2, x_{k+2}, x_{2k+2}, \ldots, x_{hk+2}$ will be encrypted by the permutation σ_2, and so on.

This is a truly algebraic idea: a complex code is broken into the direct sum of k simple codes.

If the keyword has length k, we construct, taking one letter every k, k messages, each one of which has been encoded with a single permutation σ_i.

Finally a statistical attack is used on each of the k messages found (see e.g. [8], 3.3).

This attack is possible, because a one-letter substitution code does not change the statistic of the message.

Kasiski himself proposed a method for finding the length of the key, but today we have much more effective systems, for example the Friedman index.

Despite this weakness, we observe that there are still studies and applications of Vigenère's method ([45], [11], [79], [99]).

2. The Friedman index

William Friedman, an eclectic scientist ([43]), was one of the most renowned cryptographers in history. He even studied ([83]) the famous Voynich Manuscript! He introduced the coincidence index in 1922 ([38]). This is a very fundamental idea. Given a text T written by using q different characters, the index of coincidence $I(T)$ is the probability that by taking at random two symbols in the text, they are equal. Supposing that T contains n characters and that the $i - th$ symbol appears n_i times, then $I(T)$ is given by the formula

$$I(T) = \sum_{i=1}^{q} \frac{n_i(n_i - 1)}{n(n-1)} \tag{9}$$

By calculating the average of $I(T)$ for many texts written in a given language L, we determine a coincidence index for L itself, $I(\mathrm{L})$. We call random language $\mathrm{R}(q)$ that one in which each of the q characters is randomly selected with probability $\frac{1}{q}$. Obviously $I(\mathrm{R}(q)) = \frac{1}{q}$. If we encrypt a text T with a Vigenère cipher, with key K, obtaining the ciphertext $E_K(T)$, we will observe that $I(E_K(T))$ approaches $I(\mathrm{R}(q))$ by increasing the length k of the key. This fact can be used to determine the length of the key. Let's take a couple of examples..

First of all, let's update the Vigenère cipher, so that we can encode every byte string.

Symbols are bytes, which are 8-bit numbers, ranging from 0 to 255. A text T is a string of bytes of length n. Key K is a string of bytes of length k. We then apply the method (8), where the permutations σ_i are simply cyclic shifts of the byte sequence. Coding starts by repeatedly writing K under T and then adding byte per byte modulo 256. For example, if $\mathrm{T} = (144, 90, 69, 102)$, $k = 3$ and $K = (70, 200, 240)$ then the text T encrypted with the key K is

$$E_K(T) = (144, 90, 69, 102) + (70, 200, 240, 70) \mod 256 =$$

$$(214, 34, 53, 172)$$

So from our point of view, a text is simply a finite sequence of bytes. Any file on our computer can be considered text, reading it one byte at a time. Notice that we have 256 characters, but each text has its own particular q, which is the number of distinct bytes that appear in it. Given a text T and an integer d, we define T_d

as the text that is obtained from T taking one character every d. Finally, we define $I(T,d)$ as the function $I(T,d) = I(T_d)$.

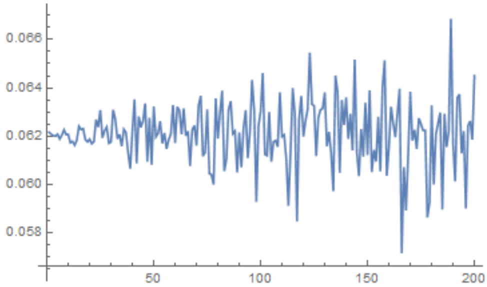
Suppose T was encrypted with a key K of length k and $C_K = E_K(T)$. Then all the characters of C_K have been encrypted with the same permutation and, in the graph of $I(C_K, d)$, as d varies, we will observe peaks corresponding to the multiples of k. We will then find the length of the key.

Let's come to the examples. The T text considered is the dante.txt file, the complete text of the Divine Comedy. It contains 573753 characters.

We calculate the coincidence index of T with the formula (9). In this case $q = 82$, because there are 82 different bytes.

We get $I(T) = 0.06217$, while the index of a random text with 82 symbols is $1/82 = 0.01219$.

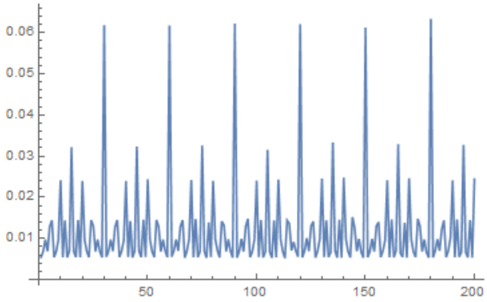
The graph of the function $I(T,d)$, with $1 \leq d \leq 200$ is



Now we encode T with Vigenère and a random key K with $k = 30$.

We have $C = E_K(T)$.

The graph of the function $I(C,d)$, with $1 \leq d \leq 200$ is



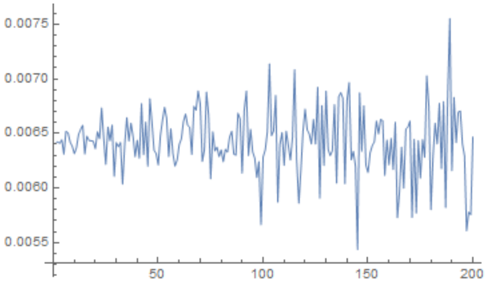
In C there are $q = 256$ distinct characters (i.e. there are all), $I(C) = 0.00590$ and $1/256 = 0.00390$.

It is clear that the highest peaks are at multiples of 30. The minor peaks correspond to the length divisors. In this way, by examining only the ciphertext C, we find out the length of the key!

Note that this technique does not require you to know in which language the text is written, it does not make direct use of the $I(\mathrm{L})$ index. It is based solely on the fact that the language used is structured, not casual. That is, it is based on the difference between $I(\mathrm{L})$ and $I(\mathrm{R}(q))$. This difference is seen in the peaks of the graph, which appear when d is a multiple of the key length. In fact, in this case, the sub-text examined has been encrypted with a mono-alphabetical substitution code, and therefore it maintains the statisics of the language L .

So let's take an example with a completely different type of file, an exe file. Let's take the gp.exe file, the executable of the beautiful computer algebra system Pari/Gp.

This is the graph of $I(gp.exe, d)$, $1 \leq d \leq 200$



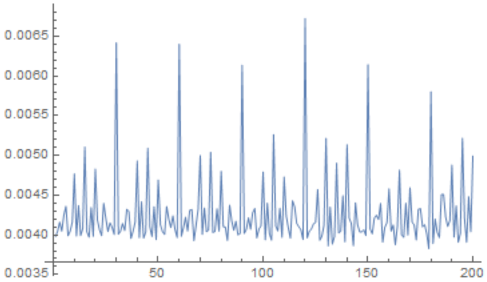
In gp.exe contains 245248 bytes, and $q = 256$.

We have $I(gp.exe) = 0.00640725$, while $I(\mathrm{R}(256)) = 0.00390625$.

As before we encode *gp.exe* with Vigenère and a random key K with $k = 30$.

We pose $S = E_K(gp.exe)$.

The graph of the function $I(S, d)$, with $1 \leq d \leq 200$ is



The index of the encoded text S, $I(S)$,dropped from the original 0.00640725 to 0.00397486. Peaks are clearly visible at multiples of 30.

Using the Friedman index, by analyzing the graph of $I(S, d)$, whoever intercepts the encrypted message S can find out if it has been encoded with a polyalphabetic substitution code, and know the length of the key, without knowing anything about the nature of the original text. Obviously, not knowing the language of the source, it will not be possible, in the decryption phase, to use a statistical attack.

As noted in [2], although substitution codes are not safe in themselves, they are used as constituent parts of other, more powerful codes. It is therefore important to thoroughly examine the vulnerability of these systems. And it is necessary to do it automatically, given the immense amount of data in circulation. Several types of algorithms are used, including genetic and compression, see [31], [2], [48], [87].

A variant of the Friedman index, called the Progress Index of Coincidence (PIC), was used in [96] to define a good fitness function for a genetic algorithm that is capable of breaking the Enigma code.

The index of coincidence is also used in fields other than cryptography. For example in [100], where a distance between human languages is introduced, and in [39], where the use of this index is proposed to determine new patterns and evolutionary signatures in DNA sequences.

The Friedman index (also known as the Kappa index) is an important research tool, and certainly deserves further study on it.

3. Enigmatic Perfection

In polyalphabetic substitution ciphers, the key must be long. The longer the key, the closer the Friedman index of the text approaches the index of the random language, and, consequently, the peaks in the graph are hardly detectable.
Furthermore, of course, the key must be unpredictable.

In Germany, a machine was produced, the Enigma machine, which transformed a plaintext into the ciphertext, changing the permutation used for each letter that was written. If the source text is formed by the characters $x_1, x_2, \ldots, x_n$ then the x_i is encoded with the permutation σ_i. The permutations are changed by rotating some disks. The permutations were repeated periodically, but the period (i.e. the length of the keyword) was very large, tens of thousands of characters. The Enigma code was broken by the collective work of two teams of researchers, resident in England and Poland. The best known of them is Alan Turing ([25]), one of the founders of computer science and complexity theory.

For the history and operation of the Enigma machine see [8] Ch. 8, [29] Ch. 8 – 9, [57] Ch. 4 – 5, [23], [98], [53].

The events that marked the breaking of the Enigma code were really interesting, even at a theoretical level and developed, among other things, a fruitful dispute between algebraists and probabilists, see [70] p. 70 – 76.

Trying to decrypt some short messages, encoded with the Enigma machine, is still a challenge today, and can be very instructive, see [90], [77], [78].

Tackling the Enigma remains an excellent benchmark for new automatic decryption methods.

To proceed further, it is useful to reduce the Vigenère method to the essential. Let us consider the alphabet formed only by the two symbols 0 and 1, the bits. A key is a string of permutations of the alphabet. In our case there are only two permutations: the identical one and the one that swaps 0 and 1. We can then forget about permutations and instead use the bit XOR, that is the sum modulo 2. The key becomes a string of bits, 0 stands for the identical permutation and 1 stands for the swap. We want the key to be long, so take it as long as the message! We want it to be unpredictable,so we choose the bits at random!

This is the OTP (One-Time Pad), the central figure of our story.

In [13] it is proved that the inventor of the OTP was Miller, who discovered it about 35 years before the two re-discoverers, Vernam and Mauborgne. Usually the OTP is called Vernam code or perfect code.

We recall the definition of the OTP code.

The message M is a n bit string, of arbitrary length n. The key K is also a n bit string.

$$M = x_1, x_2, \ldots, x_i, \ldots, x_n$$

$$k = k_1, k_2, \ldots, k_i, \ldots, k_n$$

The encrypted massage C is obtained by XORing the bits of the message with those of the key.

$$C = c_1, c_2, , \ldots, c_i, \ldots, c_n$$

where $\forall i,\ c_i = x_i + k_i \mod 2$.

Four conditions must be met for the OTP code:

a) the key must be as long as the message
b) the key must be random
c) the key must be used only once
d) the key must be kept secret.

In 1949 Claude Shannon ([92]) proved that the OTP code is (the only) perfect code. This means that if the text T has been correctly encrypted with OTP, obtaining the text in cipher C, whoever intercepts C cannot obtain any information about T, regardless of the computing power at his disposal.

The reason is this: C can come with the same probability from any text U, of the same length as T.

Even the brute force attack is not effective. If T is long n there are 2^n texts of equal length. And there are 2^n keys K. If we take all the keys one by one and calculate C XOR K we will find, in an unpredictable order, all the possible messages U, and we will never know which one was really sent!

It is clear that there is a major key management problem in the OTP. The key must not only possess the qualities a,b,c,d, but, of course, it must be shared between those who send the message and those who can legitimately receive it. Now the question arises, if A and B can share, on a secure channel, a n bit key, why don't they directly share the message M, which has the same length?

There are many ways to overcome these difficulties. OTP has actually been used in the past, in communications that required a very high degree of security ([8] p. $103 - 115$, [52] p. $714 - 731$).

Particularly interesting was the SIGSALY system, designed to encrypt telephone conversations.

With the Sigsaly machine an attempt was made to realize OTP in the transmission of voice conversations. The voice was digitized and compressed (to save bandwidth). Eventually it was represented by strings of integers between 0 and 5. A key string was added to the entry string modulo 6. The key was simply derived from a 16 inch vinyl record. Obviously, the record had to be the same for whoever spoke and who received. In the decoding phase the key was subtracted modulo 6.

SIGSALY was never broken. Turing also took an interest in it and carried out his own project to improve it ([36], Ch. 7).

SIGSALY weighed 50 tons, and the record was enough for ten minutes of broadcast. Nowadays everything has changed, thanks to technology.

We can easily use our laptop to transmit OTP encrypted texts, imitating the SIGSALY process. My message M is a string of n bits. The only problem is sharing a string of n bits (the key) with the receiver R. There are billions of terabytes available on the net! R and I download the same text from the network (or piece of music, or painting or any digital object), we perform the same preprocessing operation on it, to destroy any internal structures (for example we zip the file), we take n bits from a certain position, and we have the common key K! So I send on the network $C = M$ XOR K, and R, when it receives C, computes C XOR K.

I could also transmit to R the key K, which I produced, hiding it, by means of steganography, in an image file, or the like. We can share the address of a huge database with an algorithm that compute the sequence of the objects to be used and the starting points. This would provide a substantially unlimited key, to be used as many times as desired, always taking different parts. It is a pattern whose technical details can be bridged in many different ways. This is why there are so many articles that revolve around similar concepts.

It is not possible to quote all relevant articles, to trace a complete history of what we might call the actualization of the OTP. See these works and their bibliographies: [85], [74], [26], [102], [76].

There are very recent applications to new cryptographic methods and modern technologies, for example to Single Sign-On ([54]) and Wireless Communications ([61]).

A very interesting idea is to put together OTP and DNA. For the fundamentals of the theory see [40] and [17].

As observed in ([103]) there are at least two ways to use DNA: manipulate it directly in a laboratory with biochemical tools, or simply consider it as a code. The genetic code is a four base sequence A - adenine, C - cytosine, G - guanine, T - thymine, which can be easily converted into a binary sequence, by means of substitutions

$$A \to 00 \quad C \to 01 \quad G \to 10 \quad T \to 11$$

These binary strings are manipulated so that they can be used as keys in the OTP ([18]).

A DNA based (biochemical) method for random key generation and OTP management is presented in [103]. In [80] a one-time-pad cipher algorithm based on confusion mapping and DNA storage technology is proposed. There is also a very recent implementation in Python ([1]).

4. OTP approximations

One of the advantages of DNA encryption is that you can share keys using huge databases that are public (e.g. The National Center for Biotechnology Information).

However, in reality, as we have already noted above, we can try to use OTP easily. It is sufficient (for example) that one of us gives or sends to the other one a DVD containing a long string of bits ($\approx 10^{14}$). The string, with a particular segmentation and synchronization program, can be used as a key for many encrypted OTP communications.

Can we trust the keys taken from our disk? Let's look at the question. Shannon ([92]) showed that to achieve perfection it is necessary that the space of the keys be as large as that of the messages. Therefore 2^n keys must be available. That is, any binary string of length n must be available as a key. The single key used can be any string. What matters is that we must have chosen it, with uniform probability, in a basket that contains all the strings of length n. It is therefore not very credible that the conditions necessary for the application of Shannon's Theorem are verified.

Our dream would be this: to be able to use a short key, which can be easily shared on a secure channel, and then use it with OTP in its perfection. It seems an impossible dream, but it can be realized if we are satisfied with an approximate perfection.

In a seminal paper from 1992, Ueli M. Maurer presented a new approach in which a public source of sequences of random bits is used. Let's see in detail Maurer's method.

Suppose that user A wants to send B the n-bit message M, $M = (x_1, x_2, \ldots, x_n)$.

The publicly-accessible R is an array of independent and completely random binary random variables. R consists of m blocks of length T. The block i, with $1 \leq i \leq m$ contains the bits ($R[i, 0]$, $R[i, 1]$, ..., $R[i, T-1]$).

Now A creates a secret key Z, $Z = (Z_1, Z_2, \ldots, Z_m)$, where $0 \leq Z_h \leq T-1$ for every h. Z must be chosen from the set $(0, 1, \ldots, T-1)^m$ with uniform probability. A sends Z to B on a secure channel.

Using Z, A builds the key K, to be used with OTP.

We set $S[h] = (R[h, Z_h + j - 1] \mod T)$ with $1 \leq h \leq m$ and $j = 1, \ldots, n$.

Finally the OTP key is $K = \oplus_{h=1}^{m} S[h]$.

A sends to B the message $C = K \oplus M$. Even B can calculate K, because he knows Z and R is public, and finds $M = K \oplus C$.

We always assume that $T \gg n$. R contains $L = mT$ random bits. The binary length of the secret key Z is $\approx m \log_2 T$. Those who know Z must examine only mn of the L bits, that is a very small fraction n/T. Let's imagine that the opponent can, with the best strategy (even probabilistic), examine N bits. Mauer proves that if a certain event $\mathcal{E}$ occurs, the code is safe in Shannon's sense, i.e. the opponent cannot obtain any information about the plaintext X from the ciphertext C. The nature of the event $\mathcal{E}$ is not important, what matters is its probability, $P(\mathcal{E})$. Maurer proves that $P(\mathcal{E}) = 1 - n\delta^m$. This probability is extremely high, because $\delta = \frac{N}{L}$.

Maurer's Theorem is based on the fact that the opponent is storage-bounded, and can only examine a delta fraction of the bits of R.

Let's take an example. I want to send B a document X containing 2^{27} bits, about 100 Mb. With the classic OTP I would have to share with B a random string (the key) of 100 Mb. Instead, I apply Maurer's method assuming $m = 40$ and $T = 10^{10}$. Suppose that opponent's limit forces him to examine no more than 1/3 of R, i.e. delta $= 1/3$. I create a secret key Z, which will be about 1328 bits long, as $m \log_2(T) \approx 1328$. I share Z with B, and send the encrypted text C. By intercepting C, the opponent cannot have any information about X, with probability $1 - 2^{27}(1/3)^{40} = 1 - 10^{-11}$.

Mauer's idea was revived, modified and perfected by Rabin and others, see [28], [82], [88].

In [88] the short key dream is essentially fulfilled. The final key, which complies with Shannon's requirements, is created through an ingenious process of manufacturing intermediate keys. With an iterative method, the author shows that the relationship between key length and data can be made as small as desired, at the cost of increased computational complexity. So, surprisingly, the birth of quantum computers or other extremely powerful devices will not facilitate the breaking of the code (which is impossible, as we know) but will make the encoding of messages with OTP very fast, and the key very short!

5. The Need for Randomness

We need randomness in everyday life. Pizza or pasta tonight? We flip a coin, or do odds and evens. As Hayes says in [47], there is a real Randomness Industry. Inside each slot machine there is a special chip that continuously calculates random numbers. Immense amounts of random numbers are used every second around the world in video games, simulations, optimization algorithms, probabilistic algorithms, Monte Carlo methods and, we know, cryptography.

Any cryptographic system requires the use of keys. A sure rule is that there are no secure systems with keys that are too short, for the simple reason that, if the key is n bits long, with a brute force attack it is enough to find all 2^n keys. On the other hand, with a key of a few hundred bits we would like to encrypt messages of many thousands of bits.

We limit ourselves in this survey to stream codes, direct emanations of our OTP, which can be considered their prototype.

It is important to note that many block ciphers, for example AES, can be used as stream codes, using techniques such as Output Feedback (OFB) and others, recommended by NIST ([32], [33]).

In general, the encoding in a stream code occurs exactly as in OTP. We have a stream of binary messages $m_1, .., m_t, ..$ and keys $k_1, .., k_t ..$, we get a stream of encrypted messages $c_j = m_j \oplus k_j$.

The key stream is generated from an initial secret and, of course, random key K. If A and B want to communicate, they share K and both generate the same key stream. For reference texts see [86] and [55].

During the 10th International Conference on the Theory and Application of Cryptology and Information Security (2004), Adi Shamir gave a lecture entitled Stream Ciphers: Dead or Alive? ([91]). In his presentation, Shamir talked about a decline of stream ciphers, unlikely to be reversed in the near future. However, the author highlighted two particular areas, in which stream ciphers could have maintained priority. He said "I believe that stream ciphers will remain competitive in two types of applications: a) hardware oriented scheme with exceptionally small footprint (gates, power consumption, etc) b) software oriented scheme with exceptionally high speed"

These considerations sparked a lively discussion and in the same year (2004) eSTREAM: the ECRYPT Stream Cipher Project was launched. The competition ended in 2008. A full description of the finalists can be found in this book [84]. For a survey on stream ciphers see [50].

The research focuses on particularly fast or light stream codes.

Among the fastest codes there are Rabbit ([16]) and Salsa family ([14], [73]).

Lightweight Cryptographic Algorithms are increasingly important in the IoT. They are especially needed when dealing with small medical implants, battery-powered handheld devices, embedded systems, RFID and Sensor Networks, see Nist Internal Report [69]. A detailed study of Low Energy Stream Ciphers is here [7].

Everything we have seen requires the use of random bit strings.

6. Randomness

Let us begin by remarking that these bit (or number) sequences are produced by programs running on ordinary computers, and are therefore completely deterministic.

Calling them random seems strange. In fact they are said pseudo random. Their generators are called PRNG, Pseudo Random Number Generators.

Thousands of articles have been and will be written about them ([12]).

What do we mean by a random sequence?

Goldreich, in [44], summarizes the basic concepts of the three main theories very well.

a) In his Information Theory (1948) Shannon characterizes perfect randomness as the extreme case in which the string of symbols does not contain any redundancy, i.e. there is a maximum amount of information.
b) Solomonov (1960), Kolmogorov (1963) and Chaitin (1965), founded the second, computational theory. The complexity of a string is essentially the length of the smallest program that can generate it. In essence, if a string is truly random, a program must contain it in order to express it.
c) Blum, Goldwasser, Micali and Yao began, in the years $1982-84$, the third theory which pays attention to the actual computation. A sequence is random if we do not have computational procedures to distinguish it from a uniform distribution.

Chaitin-Kolmogorov's theory is fascinating, because it makes it possible to deal with the randomness of a single string, without resorting to any probability distribution ([22], [19], [62], [30], [93], [37]).

According to it we see that three concepts are essentially equivalent: randomness, incompressibility and unpredictability.

This is also in accordance with our intuition. A random event cannot be predictable, and in order to compress a string of bits it needs to have some regularity.

There are many efficient compression algorithms available today. The compressed text is expected to approach a random text. In this context, some compressors are studied in [56], and it is proved that arithmetic coding seems to produce perfectly random output.

If we delve into the subject, several surprises await us.

7. Incompressibility

We say that a string of n bits is c-incompressible if it cannot be compressed more than c bits. A simple counting argument ([62], p. 117) shows that there are at least $2^n - 2^{n-c} + 1$ c-incompressible strings. So there is at least one n-bit string that cannot be compressed even by one bit, at least half of the strings are 1-incompressible, the three quarts are 2-incompressible and so on.

The extreme majority of strings are very little compressible, and therefore highly random!

Can any relationship exist between the infinity of prime numbers and the incompressibility of information?

There are many different proofs of the infinity of primes. Often they are based on the fact that if the primes were finite, something would happen which is false. We have also made one [21]. If primes were finite, N would be a field!

A proof I love is due to Chaitin (see [19] p. 361). If primes were finite, almost everything would be compressible.

Suppose there are only k primes, $p_1, p_2, ..., p_k$. Given an integer $N > 1$, by the unique factorization theorem, we will have

$$N = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$$

Clearly we have $k \leq \log_2 N$ and $\forall i,\ e_i \leq \log_2 N$.

The number N is identified precisely by the string of exponents $e_1, ..., e_k$. These exponents are integers $\leq \log_2 N$ and can therefore be expressed (each) by $\log_2 \log_2 N$ bits.

In conclusion, every integer N can be expressed by

$$k \log_2 \log_2 N \tag{10}$$

bits.

Let us take a string M containing m bits. This uniquely identifies an integer N of m bits, of order 2^m.

By (10) N, and hence M, is determined by a string of $k \log_2 m$ bits.

There exists m_0 such that

$$\forall m > m_0 \quad m > k \log_2 m$$

From this we deduce that all sufficiently long bit strings are compressible!

But we know very well that this is not true.

What if we insist on compressing everything?

Of course this is possible, if we accept that we cannot go back, that we lose information. For example, given a message M of n bits, with n large, we can decide to take the first 128 bits of M. This does not seem very useful.

However, it would be convenient to create, for each message M of length n, a kind of 128-bit fingerprint, which would uniquely identify it. This is clearly impossible, if $n > 128$, but hashing techniques try to come close to this dream.

Hash functions are very important in cryptography. A hash function H transforms a message M of any length n into a string of fixed length m. Typically m is 128 or 256.

A cryptographic hash function H must satisfy two conditions:

- given $y = H(x)$ it must be computationally difficult to find x
- it must be computationally difficult to determine x and x' such that $H(x) = H(x')$

Two other properties, not easily formalized, are required in practice.

The hash $H(x)$ must appear random (here we go again) even if H is perfectly deterministic. It is also required that H is sensitive to initial conditions. In the sense that, if we change a single bit in x, about half of the bits of $H(x)$ change.

Hash functions are public and intensively used, see Ch. 9 of [34], Ch. 7 of [49].

Hash functions have a thousand uses, ranging from database indexing to electronic signature (it is much faster to sign $H(M)$ than M). They are used in many cryptographic protocols, such as bit commitment and password management. Precisely in this last area we have patented a system that contains a rather interesting hash function,based on the Chinese Remainder Theorem ([81]).

As is known, bitcoin was introduced by Satoshi Nakamoto in 2008 ([75]). Hash functions are the real engine of the so-called mining that bitcoin uses: one must find an x such that $H(x) \leq t$, where t is a target 256 bits string, see Ch. 2 of [27] and 10.5 of [49]. Thanks to bitcoins, through hash functions, randomness passes directly into the economy, and earning capital becomes a worldwide lottery. Bitcoin mining involves an enormous consumption of energy. For an in-depth analysis of the impact of bitcoins on the economy and the environment, see [5].

8. Unpredictability

We say that a sequence of bits $S = (b_n)$ is unpredictable if it passes the next bit test ([58], §6.1), i.e. if there is no polynomial time algorithm which, receiving the first $n-1$ bits of S as input, returns the nth bit with probability $p > 1/2$.

In a truly masterful article ([15]) L. Blum, M.Blum and M. Shub examined the predictability of two PRNG.

One of them is the quadratic generator.

Let $N = pq$, where p, q are primes congruent to 3 modulo 4. We choose an integer x_0, the seed.

Then we produce the sequence

$$x_{n+1} = x_n^2 \mod N$$

The random bit sequence generated is $b_n = x_n \mod 2$.

The authors prove that the sequence b_n is unpredictable, as long as you don't know how to factor N.

In their article BBS studied a second generator, besides the quadratic one, the $1/P$ generator, which is truly remarkable.

P is a prime number, and b is a basis. If b generates the multiplicative group $\mathbb{Z}_P^*$, the expression of $1/P$ in base b has period $P-1$. We can thus obtain very large periods, since today it is easy to find prime numbers with hundreds of digits.

If b generates $\mathbb{Z}_P^*$, you get long strings of $b-$digits, which pass the statistical tests with good results.

Surprisingly BBS show that the $1/P$ generator is easily predictable: in fact it is sufficient to know $2l$ consecutive digits (where l is the length of P in the base b) to deduce the entire period. The method is based on continued fractions.

In general, the length of the period alone does not guarantee security. For example the Mersenne Twister has a huge period,

$2^{19937} - 1$ but, as stated by the authors themselves (see ([67])), it is not suitable for cryptographic purposes: indeed the output of the algorithm becomes a linear recurring sequence by applying a simple linear transformation.

In prediction problems, it is natural to use Soft Computing or Artificial Intelligence techniques.

A few years ago, with Mario Giacobini and Pierre Liardet ([20]), we studied the prediction capabilities of Finite State Automata (FSA), evolving populations of automata using Genetic Algorithms.

The underlying idea was to use the evolutive ability of prediction of the algorithm to get measures of the randomness of the sequence. Among other things we found that

- the evolved prediction skills are in inverse proportion to the period length of the considered sequence;

- the evolution of FSAs prediction skills seems to be directly linked to the linear complexity of the sequences considered.

These last two conclusions make us hope that the evolution of FSAs could be used as a measure of the randomness character of a binary string.

In ([95]) T. Smith considers automata of different types for the prediction of infinite strings with various types of periodicity.

These machines, called predictors, can have multiple heads that at each step move along the assigned infinite sequence, read the symbol, change state and make a guess about the next symbol before it appears. This interesting paper describes new prediction algorithms for the classes of purely periodic, ultimately periodic, and multilinear words.

The sequence of the digits of Pi (in any basis) is believed to be random (see §10). Therefore, the results of Fa and Wang are surprising. In ([35]) Fenglei Fa and Ge Wang use Neural Networks to make predictions on the bits of Pi. Their neural networks predict the next bit (of 6 bit strings) with probability $> 1/2$.

The authors also apply the method to strings generated by PRNG. They conclude that neural networks, even very simple ones, can extract useful information for prediction from the data (if we are not in the presence of maximum entropy, or total disorder).

In the remarkable article by Taketa et al. [97] it is observed that neural networks learn to predict the next bit with a particular sensitivity to the linear complexity of the sequence.

The field of machine learning in pseudorandom sequence prediction is just beginning and is truly fascinating.

9. Pseudo Random Number Generators

Most of us would like to have a TRNG (Truly Random Number Generator) available. On the net there are several sources, see www.random.org. A now classic is www.fourmilab.ch/hotbits: HotBits are generated by timing successive pairs of radioactive decays detected by a Geiger-Müller tube interfaced to a computer.

This may not necessarily be the best choice. As Donald Knuth observes in [59] p. 145, "a truly random sequence will exhibit local nonrandomness"; there will certainly be, for example, sequences consisting of a million consecutive zeros. OTP would send a million bits as plain text!

Here, as we said, we make do with PRNGs.

At the beginning of the chapter on random numbers (now a classic of the subject, recommended to all), Knuth recalls a fact from his youth. He built a very complicated algorithm to generate random sequences. Unfortunately when it was activated on the computer, sequences were observed that were repeated with a very short period! This is Knuth's ([59], p.5) conclusion:

“The moral of this story is that random numbers should not be generated with a method chosen at random. Some theory should be used. ”

Theory, examples and applications can be found here: [34] Ch. 12, [51] and [58].

[51] is particularly suitable for engineers and programmers, and also contains valuable information on the actual use of many programs.

Also in [58] there is a particular attention to programming. It is a very well written text, rich in content. I found enjoyable the use of the Monty Hall Dilemma ([72], Ch.3) which is proposed, with C-code and concrete examples, as a test of randomness!

Cryptography requires random sequences of a particular type: see the overview [66] which states that one of the most important characteristics is that of unpredictability. In the survey [3], in addition to these problems, quasi-random numbers are also considered.

In practice it is not possible to know a priori the qualities of a PRNG, it must be subjected to batteries of randomness tests. Many of them are currently in use, we quote Ent, Diehard, NIST, TestU01. See [58] Ch. 4, [51] Ch. $8-11$, the important article of Shen [94] and the recent ACM recommendations [63].

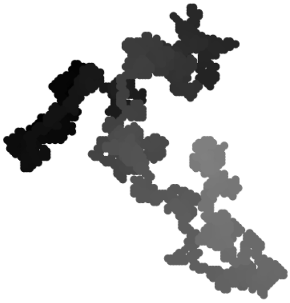
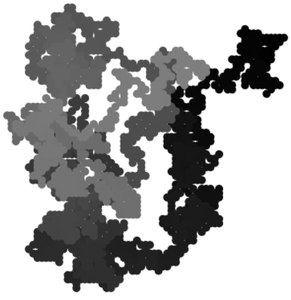
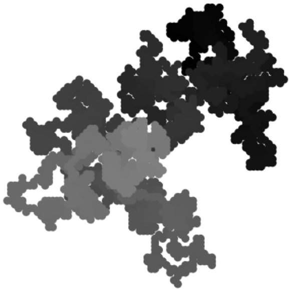
10. Irrational Randomness

As we saw in §4, talking about DNA and the ideas of Mauer and others, OTP can be realized starting from a large and public database of random bits (or numbers). If really big, we wouldn't even need to change it. Users only need to share the access points and formatting algorithms. This is a great idea, and it is revived periodically, in ever new ways.

In [24] the author wonders if it would be better to extract rather than expand. The classical OTP is revisited, then the expansion paradigm is compared to the idea of extraction. Users A and B only need to share a "mother pad". Gualtieri in [46] suggests using the sequence of the digits of Pi. Naturally then each author proposes his own method to manage the common source of randomness.

But are Pi's digits really random? Marsaglia's authoritative opinion is positive, see [64] and [65]. There has been much debate on the question, see for example [41], [6] and [42]. Empirically π seems to pass all tests of randomness ([71]).

We try to understand how random π is also in a qualitative way, mainly visual, look at the beautiful Walking on real numbers ([4]). π is indistinguishable from the sequences generated by PRNG even in the fractals produced by Chaos Games ([89]). It seems that the first to walk on the π digits was Venn ([101]). Pi was developed in base 8, and each digit $0, .., 7$ was associated with a direction. Of course numbers other than π can be used, we have uncountable choices! Venn's idea is used below to see the first 4000 digits of π, e and $\cos(1)$ (left to right)


,

,


I believe that visualizing the numbers in a meaningful way is a very important project to pursue, we are only at the beginning of a great adventure!

To conclude, we can say that the Short Key Dream can also be realized in a different way: just whisper in your friend's ear (assuming the necessary technical details have been shared) "cos(5)".

CrypTOrino 2021
ISBN 000-00-000-0000-0
DOI 10.4399/000000000000023
pag. 117–124 (xx 9999)

Bibliografia

[1] Abdelghany Faten M., Rahouma Kamel H., Hassan Yahia B., Mahdy Lamiaa N., Design and Implementation of a Web-DNA-Based Security Provider Using Python, International Conference on Advanced Machine Learning Technologies and Applications (AMLTA), (2021).

[2] Alkazaz N. R., Irvine S. A., Teahan W. J., An automatic cryptanalysis of simple substitution ciphers using compression, Information Security, Information Security Journal: A Global Perspective, Vol. 27, p. 57 – 75 (2018).

[3] Aljahdali A. O., Random Number Generators Survey, International Journal of Computer Science and Information Security (IJCSIS), Vol. 18, No. 10, p. 14 – 22 (2020).

[4] Aragón Artacho, F.J., Bailey, D.H., Borwein, J.M. et al., Walking on Real Numbers, Math Intelligencer Vol. 35, p. 42 – 60 (2013).

[5] Badea M., Mungiu-Pupăzan M. C., The Economic and Environmental Impact of Bitcoin, in IEEE Access, Vol. 9, p. 48091 – 48104 (2021).

[6] Bailey D. H., Borwein J. M., Brent R. P., Reisi M., Reproducibility in Computational Science: A Case Study: Randomness of the Digits of Pi, Experimental Mathematics, Vol. 26, 298 – 305 (2017).

[7] Banik S. et al., Towards Low Energy Stream Ciphers, IACR Transactions on Symmetric Cryptology, Vol. 2018, No. 2, p. 1 – 19 (2018).

[8] Bauer C. P., Secret History: The Story of Cryptology, CRC Press (2013).

[9] Bauer C. P., Unsolved!, Princeton University Press (2017).

[10] Bauer F. L., Decrypted Secrets, Methods and Maxims of Cryptology, Springer (2007).

[11] Bazeer Ahamed B., Krishnamoorthy M., SMS Encryption and Decryption Using Modified Vigener Cipher Algorithm, Journal of the Operations Research Society of China (2020).

[12] Beebe N. F. H., A Bibliography of Pseudorandom Number Generation, Sampling, Selection, Distribution, and Testing, University of Utah, Department of Mathematics (2019).

[13] Bellovin S. M., Frank Miller: Inventor of the One-Time Pad, Cryptologia, Vol. 35, p. 203 – 222 (2011).

[14] Bernstein D. J., The Salsa20 family of stream ciphers, Lecture Notes in Computer Science, Vol. 4986, p. 84 – 97 (2008).

[15] Blum L., Blum M., Shub M., A Simple Unpredictable Pseudo-Random Number Generator, SIAM J. Comput., Vol. 15, p. 364−383 (1986).

[16] Boesgaard M., Vesterager M., Pedersen T., ChristiansenJ., Scavenius O., Rabbit: A New High-Performance Stream Cipher, T. Johansson (Ed.): FSE 2003, LNCS 2887, p. 307 – 329 (2003).

[17] Borda M. E., Tornea O., DNA Secret Writing Techniques, 8th International Conference on Communications, p. 451 – 456 (2010).

[18] Borda M. E., Tornea O., Terebes, R., Malutan, R.,New DNA Based Random Sequence Generation and OTP Encryption Systems for Transmission and Storage, Proceedings of 2013 International Conference on Security for Information Technology and Communications (SECITC) (2013).

[19] Calude C. S., Information and randomness: an algorithmic perspective, Springer Science & Business Media (2002).

[20] Cerruti U., Giacobini M., Liardet P., Prediction of Binary Sequences by Evolving Finite State Machines, In: Collet P., Fonlupt C., Hao JK., Lutton E., Schoenauer M. (eds) Artificial Evolution. EA 2001. Lecture Notes in Computer Science, vol 2310. Springer, Berlin, Heidelberg (2002).

[21] Cerruti U., Murru M., If the Primes are Finite, Then All of Them Divide the Number One, American Mathematical Monthly, Vol.124, p. 969 (2017).

[22] Chaitin G., Algorithmic Information Theory, Cambridge University Press (1990).

[23] Christensen C., Review of IEEE Mileston eAward to the Polish Cipher Bureau for The First Breaking of Enigma Code, Cryptologia, Vol. 39, p. 178 – 193 (2015).

[24] Coluccia A., Rethinking Stream Ciphers: Can Extracting be Better than Expanding?, Wireless Pers Commun Vol. 73, p. 77−94 (2013).

[25] Cooper S. B., Van Leeuwen J. (editors), Alan Turing, His Work and Impact, Elsevier (2013).

[26] David R., Măluţan R., Borda M., TLS protocol: Improving using ElGamal elliptic curves and one-time-pad, 11th International Symposium on Electronics and Telecommunications (ISETC), p. 1 – 4 (2014).

[27] Dhillon V., Metcalf D., Hooper M., Blockchain Enabled Applications, Apress, Berkeley, CA, Second Edition (2021).

[28] Ding Y. Z., Rabin M. O., Hyper-Encryption and Everlasting Security, Annual Symposium on Theoretical Aspects of Computer Science, STACS 2002 (LNCS 2285), p. $1 - 26$.

[29] Dooley J. F., History of Cryptography and Cryptanalysis, Springer (2018).

[30] Downey R. G., Hirschfeldt D. R., Algorithmic Randomness and Complexity, Springer Science & Business Media (2010).

[31] Dureha A., Kaur A., A Generic Genetic Algorithm to Automate an Attack on Classical Ciphers, International Journal of Computer Applications, Vol. 64, p. $20 - 25$ (2013).

[32] Dworkin M., Recommendation for Block 2001 Edition Cipher Modes of Operation, Methods and Techniques, NIST Special Publication $800 - 38A$ Recommendation for Block 2001 Edition.

[33] Dworkin M., Recommendation for Block Cipher Modes of Operation: Three Variants of Ciphertext Stealing for CBC Mode, NIST Special Publication $SP800 - 38A$ Addendum Recommendation for Block 2010 Edition.

[34] Easttom W., Modern Cryptography, Springer (2021).

[35] Fan F., Wang G., Learning from Pseudo-Randomness With an Artificial Neural Network - Does God Play Pseudo-Dice? https://ieeexplore.ieee.org/document/8350369IEEE Access (2018).
ì

[36] Floyd J., Bokulich A. (editors), Philosophical Explorations of the Legacy of Alan Turing, Springer (2017).

[37] Franklin J. N., Porter C. P. (editors), Algorithmic Randomness: Progress and Prospects, Cambridge University Press, Lecture Notes in Logic Vol. 50 (2020).

[38] Friedman W., The Index of Coincidence and its Applications in Cryptography, Riverbank Laboratories, Geneva (1922).

[39] Gagniuc P., Cristea P. D., Tuduce R., Ionescu-Tîrgovişte C., Gavrila L., Dna patterns and evolutionary signatures obtained through Kappa Index of Coincidence, Rev. Roum. Sci. Techn. Électrotechn. et Énerg., Vol. 57, p. $100 - 109$ (2012).

[40] Gahani A., LaBean T., Raif J., DNA-based Cryptography, in Aspects of Molecular Computing, Lecture Notes in Computer Science 2950, p. $167 - 188$ (2005).

[41] Ganz R. E., The Decimal Expansion of π Is Not Statistically Random, Experimental Mathematics, Vol. 23, p. $99 - 104$ (2014).

[42] Ganz R. E., Reply to Reproducibility in Computational Science: A Case Study: Randomness of the Digits of Pi? [Bailey et al. 17], Experimental Mathematics, Vol. 26, p. 303 – 307 (2017).

[43] Goldman I. L., William Friedman, Geneticist Turned Cryptographer, Genetics, Vol. 206, p. 1 – 8 (2017).

[44] Goldreich O., On the Impact of Cryptography on Complexity Theory, Department of Computer Science, Weizmann Institute of Science (2019).

[45] Grošek O., Antal E., Fabšič T., Remarks on breaking the Vigenère autokey cipher, Cryptologia, Vol. 43, p. 486 – 496, (2019).

[46] Gualtieri D., One-Time Pads from the Digits of π https://arxiv.org/abs/2103.08783 (2021).

[47] Hayes B., Computing Science: Randomness as a Resource, American Scientist, Vol. 89, p. 300 – 304, (2001)arXiv:1907.03251v1 (2019).

[48] Hilton R., Automated cryptanalysis of monoalphabetic substitution ciphers using stochastic optimization algorithms, PhD thesis, Department of Computer Science and Engineering, University of Colorado, Denver (2012).

[49] Hwang S. O., Kim I., Lee W. K., Modern Cryptography with Proof Techniques and Implementations, CRC Press (2021).

[50] Jiao L., Hao Y. L., Feng D. G., Stream cipher designs: a review, SCIENCE CHINA Information Sciences, Vol. 63, Issue 3 (2020).

[51] Johnston. D., Random Number Generators-Principles and Practices, Walter de Gruyter (2018).

[52] Khan D., The Codebrakers: The Story of Secret Writings, The Macmillan Company (1967).

[53] Kenyon D., Weierud F., Enigma G: The counter Enigma, Cryptologia, Vol. 44, p. 385 – 420 (2020).

[54] Kihara M., Iriyama, S., Security and Performance of Single Sign-On Based on One-Time Pad Algorithm, Cryptography, Vol. 4, 31 pages, doi:10.3390/cryptography4020016 (2020).

[55] Klein A., Stream Ciphers, Springer (2013).

[56] Klein S. T., Shapira D., On the Randomness of Compressed Data, Information 11(4), 196 (2020).

[57] Klima R., Sigmon N., Cryptology Classical and Modern, CRC Press (2019).

[58] Kneusel R. T., Random Numbers and Computers, Springer International Publishing AG, part of Springer Nature (2018).

[59] Knuth D. E.,The art of computer programming, volume 2: Seminumerical algorithms, Addison-Wesley, Second Edition (1981).

[60] Lewand R. E., Cryptological Mathematics, The Mathematical Association of America (2000).

[61] Li G., Zhang Z., Zhang J., Hu A., Encrypting Wireless Communications On the Fly Using One-Time Pad and Key Generation, IEEE Internet of Things Journal, Vol. 8, p. 357 – 369 (2021).

[62] Li M., Vitáni P., An Introduction to Kolmogorov Complexity and Its Applications, Springer (2008).

[63] Luengo E. A., Villalba L. J. G., Recommendations on Statistical Randomness Test Batteries for Cryptographic Purposes, ACM Comput. Surv. 54, 4, Article 80 (2021), 34 pages.

[64] Marsaglia G., On the Randomness of Pi and Other Decimal Expansions, InterStat (2005).

[65] Marsaglia G., Refutation of claims such as "Pi is less random than we thought ", InterStat (2006).

[66] Marton K., Suciu A., Ignat I., Randomness in Digital Cryptography: A Survey, Romanian Journal of Information Science and Technology, Vol. 13, p. 219 – 240, (2010).

[67] Matsumoto M., Nishimura T. Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator, ACM Transactions on Modeling and Computer Simulation, Vol. 8, p. 3 – 30 (1988).

[68] Mauer U. M., Conditionally-Perfect Secrecy and a Provably-Secure Randomized Cipher, J. Cryptology, Vol. 5, p. 53 – 66 (1992).

[69] McKay K. A., Bassham L., Turan M. S., Mouha N., Report on Lightweight Cryptography, NIST Internal Report 8114 (2017).

[70] Megyesi B.(editor), Proceedings of the 3^{rd} International Conference on Historical Cryptology HistoCrypt 2020, NEALT Proceedings series 44, Linköping Electronic Conference Proceedings, No. 171 (2020).

[71] Mitsui T., The Number π as a Pseudo-Random Number Generator, The Science and engineering review of Doshisha University, Vol. 49, p. 160 – 168 (2008).

[72] Mlodinow M., The Drunkard's Walk: How Randomness Rules Our Lives, Vintage (2009).

[73] Muhammad F., Ahendyarti C., Masjudin, Chacha stream cipher implementation for network security in wireless sensor network, Broad Exposure to Science and Technology 2019 (BEST2019), IOP

Conference Series: Materials Science and Engineering Vol. 673, p. 487 – 492 (2019).

[74] Nagaraj N., Vaidya V., Vaidya P. G., Re-visiting the One-Time Pad, International Journal of Network Security, Vol.6, p. 94 – 102 (2008).

[75] Nakamoto S., Bitcoin: A Peer-to-Peer Electronic Cash System.

[76] Omolara A. E., Jantan A., Abiodun O. I., Arshad H. An Enhanced Practical Difficulty of One-Time Pad Algorithm Resolving the Key Management and Distribution Problem, Proceedings of the International MultiConference of Engineers and Computer Scientists (2018) Vol I.

[77] Ostwald O., Weierud A., History and Modern Cryptanalysis of Enigma's Pluggable Reflector, Cryptologia, Vol. 40, p. 70 – 91 (2016).

[78] Ostwald O., Weierud A., Modern breaking of Enigma ciphertexts, Cryptologia, Vol. 41, p. 395 – 421 (2017).

[79] Park S., Kim J., Cho K., Yum D. H. Finding the key length of a Vigenère cipher: How to improve the twist algorithm, Cryptologia, Vol. 44, p. 197 – 204 (2020).

[80] Peng W., Cui S., Song C., One-timepad cipher algorithm based on confusion mapping and DNA storage technology,PLoS ONE 16(1):e0245506 (2021).

[81] Perna A., Abrate M., Barbero S., Cerruti U., Murru N., Method for the management of virtual objects corresponding to real objects, corresponding system and computer program product, https://patents.justia.com/patent/20170371950.

[82] Rabin M. O., Provably Unbreakable Hyper-Encryption In the Limited Access Model, IEEE Information Theory Workshop on Theory and Practice in Information-Theoretic Security, 34 – 37 (2005).

[83] Reeds J., William F. Friedman's Transcription of the Voynich Manuscript, Cryptologia, Volume 19, p. 1 – 23, (1995).

[84] Robshaw M., Billet O. (editors), New Stream Cipher Designs, Lecture Notes in Computer Science, Vol. 4986 (2008).

[85] Rubin F., One Time Pad Cryptography, Cryptologia, Vol. 20, p. 359 – 364 (1996).

[86] Rueppel R. A. Analysis and Design of Stream Ciphers, Springer (1986).

[87] Sabonchi A. K. S., Akay B., A Binomial Crossover Based Artificial Bee Colony Algorithm for Cryptanalysis of Polyalphabetic Cipher, Tehnički vjesnik, Vol. 27, p. 1825 – 1835 (2020).

[88] Sadjadpour H. R., On the Shannon Perfect Secrecy Result, 14th International Conference on Signal Processing and Communication Systems (ICSPCS), 1 – 8, (2020).

[89] Salau T. A. O,, Ajide O. O., Comparative Analyses of Iteratively Generated Fractals: The Cases of Two ?Chaos Game? Options, American Journal of Engineering Research, Vol. 6, p. 174 – 178 (2017).

[90] Schrödel T., Breaking Short Vigenère Ciphers, Cryptologia, Vol. 32, p. 334 – 347 (2008).

[91] Shamir A., Stream Ciphers: Dead or Alive?, In: Lee P.J. (eds) Advances in Cryptology - ASIACRYPT (2004). Lecture Notes in Computer Science, Vol. 3329. p. 78. Springer, Berlin, Heidelberg.

[92] Shannon C., Communication Theory of Secrecy Systems, Bell System Technical Journal, Vol. 28, p. 656 – 715 (1949).

[93] Shen, A., Uspensky, V. A., Vereshchagin, N., Kolmogorov complexity and algorithmic randomness, American Mathematical Soc., Vol. 220 (2017).

[94] Shen A., Randomness Tests: Theory and Practice, in Fields of Logic and Computation III, LNCS 12180, p. 258 – 290 (2020).

[95] Smith T., Prediction of infinite words with automata, Theory of Computing Systems, Vol. 62, p. 653 – 681 (2018).

[96] Sommervoll Ȧ. Ȧ., Nilsen L., Genetic algorithm attack on Enigma's plugboard, Cryptologia, DOI: 10.1080/01611194.2020.1721617, p. 1 – 33 (2020).

[97] Taketa Y. et al., Mutual Relationship between the Neural Network Model and Linear Complexity for Pseudorandom Binary Number Sequence, Seventh International Symposium on Computing and Networking Workshops (CANDARW), p.394 – 400 (2019).

[98] Thimbleby H. Human factors and missed solutions to Enigma design weaknesses, Cryptologia, Vol. 40, p. 177 – 202 (2016).

[99] Uniyal N., Dobhal G., Rawat A., Sikander A., A Novel Encryption Approach Based on Vigenère Cipher for Secure Data Communication, Wireless Personal Communications (2021).

[100] Vera V. J., Sánchez Ávila C., Graphemic-phonetic diachronic linguistic invariance of the frequency and of the Index of Coincidence as cryptanalytic tools, PLoS ONE 14 (3): e0213710 (2019).

[101] Verbugt L. M., The First Random Walk: A Note on John Venn?s Graph, The Mathematical Intelligencer, Vol. 42, p. 41 – 45 (2020).

[102] Voborník P., Migration of the perfect cipher to the current computing environment, WSEAS Transactions on Information Science

and Applications, Vol. 11, p. 196 – 203 (2014).

[103] Zhang Y., Liu X., Sun M., DNA based random key generation and management for OTP encryption, Biosystems, Vol 120, p. 51 – 63 (2017).

Università di Torino
umberto.cerruti@unito.it

TITOLO SERIE

Book Series

1. Autore
   Titolo
   isbn 978-88-255-xxxx-x, formato 17 × 24 cm, xxx pagine, xx euro

Compilato il 18 giugno 2024, ore 07:33
con il sistema tipografico LaTeX 2ϵ

Finito di stampare nel mese di xx del 9999
dalla tipografia «System Graphic S.r.l.»
00134 Roma – via di Torre Sant'Anastasia, 61
per conto della «Gioacchino Onorati editore S.r.l. – unipersonale» di Canterano (RM)