Meeting the Challenges of Critical and Extreme Dependability and Security

Paulo Esteves-Verissimo, Marcus Völp, Jérémie Decouchant, Vincent Rahli, Francisco Rocha
Critical and Extreme Security and Dependability Group (CritiX)
Interdisciplinary Centre for Security, Reliability and Trust (SnT)
University of Luxembourg
L-2721 Luxembourg
Email: <name>.<surname>@uni.lu

***Abstract*—The world is becoming an immense critical information infrastructure, with the fast and increasing entanglement of utilities, telecommunications, Internet, cloud, and the emerging IoT tissue. This may create enormous opportunities, but also brings about similarly extreme security and dependability risks. We predict an increase in very sophisticated targeted attacks, or advanced persistent threats (APT), and claim that this calls for expanding the frontier of security and dependability methods and techniques used in our current CII. Extreme threats require extreme defenses: we propose resilience as a unifying paradigm to endow systems with the capability of dynamically and automatically handling extreme adversary power, and sustaining perpetual and unattended operation. In this position paper, we present this vision and describe our methodology, as well as the assurance arguments we make for the ultra-resilient components and protocols they enable, illustrated with case studies in progress.**

***Index Terms*—Advanced and persistent threats, fault and intrusion tolerance, extreme computing**

I. Introduction

Information and Communication Technology (ICT) became so important in our lives that a great deal of society's stakes is today placed on the cyberspace. The pillars of this new environment are *critical information infrastructures (CII)*, increasingly considered a key factor of competitiveness of modern societies. Generally designating the computerized and networked part of physical infrastructures — such as energy, telecom, or transportation — and a relevant example of cyber-physical systems (CPS), they have been complemented over the past few years with a set of emerging computer-based CII. Increasingly relying on the Internet-Cloud complex, these infrastructures support critical assets like: the finance or public administration systems; social networks, whose societal role has been more than confirmed in recent years; health and biomedical systems like e-biobanks, whose privacy sensitiveness became an issue, with the emerging trend of massive DNA sequencing; and finally, the emerging connected, cooperating, and autonomous car ecosystems, which will soon become another relevant CII.

This work is in part supported by SnT - University of Luxembourg and Fonds National de la Recherche Luxembourg through PEARL grant FNR/P14/8149128.

We predict a fast and increasing entanglement of classical critical utility information infrastructures (electrical, gas, water, etc.) with telecommunications, Internet and cloud infrastructures, they themselves CIIs as explained above. The scenario will be enriched by an Internet-of-Things (IoT) tissue, whose explosive growth in the next few years is predicted almost unanimously by many sources. A good example of why this is inevitable is the advent of smart-grids for energy, not to mention smart homes or assisted living. Both scenarios will converge into one and only one: extremely large-scale and extremely complex computer and network systems, where classical computing devices coexist with embedded devices (many of them mobile), in a practically seamless manner; these devices will be highly programmable and dynamic; information processing ("IT") will coexist with real-time control ("SCADA"); computer-caused failures may be physical as well as virtual.

This scenario may create enormous opportunities, but also brings about similarly *extreme* security and dependability risks which, if not mastered by design, may have very negative impact on the profoundly ICT-dependent society we envision. It is recognised in the cybersecurity strategies of several countries that threats to critical information infrastructures are to be feared. The value of the assets at stake in this hugely interconnected and virtualised world is formidable and, in consequence, is attracting the attention of organised crime and cyber-terrorism, cyber-hacktivism organisations or militias, and nation-state armies or agencies.

The key focal problem points that describe this situation are: sheer *insufficient resilience of critical information infrastructures (CII)* such as core networks, data centers and power grids, against extreme levels of faults and intrusions, especially advanced *persistent threats or targeted attacks*; *new risks of computer-borne physical damage*, derived from the interconnection of telco and Internet systems with cyber-physical systems, amplified by the ever-increasing use of IoT components, networked embedded gadgets and mobile devices; organization of societies around pervasive e-xxx services and social networks based on the Internet-Cloud complex, which became de-facto critical to society, but exhibit a *security and*

978-1-5090-5652-1/17 $31.00 © 2017 IEEE
DOI 10.1109/PRDC.2017.21

dependability deficit.

We will continue to see a combination of generic security attacks and intrusions and accidental failures and disasters, exhibiting low to average severity and requiring moderate skills, mostly with overt process and effects, where severity is linear with scale (number of targets affected, e.g. DDoS). However, the new scenario we laid down above will bring in new categories of threats, especially in the malicious domain. We will witness an increase in very sophisticated targeted attacks, or advanced persistent threats (APT), perpetrated by highly skilled, motivated and highly resourced adversaries (0-day vulnerabilities, subverted hardware, interception), against selected targets, in a manner proportionate to their value. The relatively small scale and focus allows careful planning, and the projection of quite elevated power onto the targets, exhibiting high to very high severity and possibly leading to extensive damage, virtual and/or physical, including human. The preparation phase (and sometimes the execution) is normally stealth.

Our point is confirmed by recent cybersecurity incidents involving state-sponsored cyberweapons (e.g., Stuxnet, APT-1, etc.), alerting to the fact that highly-skilled adversaries are able to penetrate practically any system, even those classified as highly secure, if they are provided with enough resources and the right tools. The type of tools used for these attacks typically exploit zero-day vulnerabilities, which allow part of the invasion to go unnoticed until enough strength is gathered for the main attack.

The considerable financial and human resources that powerful organisations can gather raises justifiable concerns regarding the security of critical information infrastructures. The type of software vulnerabilities these advanced cyberweapons depend on are rare and hard to find but with the correct resources they are within reach. There are two important facts to remember that clearly support the concerns surrounding this threat. First, modern software systems tend to have large code bases (e.g., the Linux kernel beneath Android has approximately 15 million lines of code while a top-of-the-range car has close to 100 million lines of code [1]). Second, the number of software vulnerabilities grows with the number of lines of code [2], [3]. These facts place advanced cyberweapons at the forefront of persistent threats to security sensitive computer system.

Can traditional, industry grade measures like: intrusion prevention; intrusion detection and largely manual remediation; static, pre-defined and thus brittle policies and mechanisms — cope with these threats? Extreme threats require extreme defenses. We argue for the need to expand the frontier of security and dependability methods and techniques used in our current CII, to embrace what we might call *Extreme Computing - computer science and engineering pushed to the extremes of functional and non-functional properties of systems*[1]. Since modern systems require functionality, which adds complexity and lines of code, it is of paramount importance that modern security solutions are capable of enforcing a system's security requirements under extreme situations. We claim that these new categories of threats, especially malicious, require a quantum leap in the paradigms and techniques we have been using so far. In short:

- A comprehensive approach to both accidental and malicious threats, and from first principles: "building defence in", and not bolting it on, is more than ever needed.
- Providing protection in an incremental way, automatically adapting to a dynamic range of threat severity, allowing systems to be both efficient in normal times, and able to defeat extreme adversary power.
- Sustaining perpetual and unattended operation in a systematic and automatic way, in the presence of continued and persistent threats.

An increasing number of researchers have in fact been investigating along these lines over the past few years, developing powerful and innovative automatic security and dependability techniques combining fundamental paradigms, like fault and intrusion tolerance (Crash or Byzantine Fault Tolerance, CFT/BFT), secret sharing and secure multi-party computation, homomorphic encryption, erasure coding and dispersion, self-healing and diversity mechanisms. The past European Network of Excellence ReSIST is a good example of a concerted effort [4]. Projects on power grid critical infrastructures have pioneered experiments on the application of resilient technologies to real settings [5], [6]. This vision goes in line with a recent statement by NSA director and Chief of US Cyber Command, argueing in favor of cyber-resilience [7].

In this position paper, we sketch the CritiX approach and methodology to advance the frontier of research in security and dependability, preparing critical information infrastructures against the extreme threats of today and tomorrow. We strive for automated security in the presence of advanced and persistent threats and despite attacks mounted by highly skilled and well equipped adversaries. Our goal is to run systems unattended, keeping them operational and the data they process secure, even if parts of the system are already compromised. For that, we envision a widespread use of Byzantine fault and intrusion tolerant (CFT/BFT) algorithms and rejuvenation techniques, as a baseline of resilient security solutions capable of surviving continuous attacks.

However, despite an already significant body of research in the area, it is necessary to continue adapting and enhancing resilience mechanisms and protocols to the stringent requirements found in critical information infrastructures.

We describe our general, divide-and-conquer based methodology to beat these extreme threats, in Sec. II, illustrate BFT-based automated security and dependability in Sec. III and

[1]Expanding its scope from the initial meaning of grids, BigData, or HPC, towards the direction of any techniques and paradigms exploring the notion of "limits" or "boundaries" in several facets of computing, not only performance-oriented, not only functional.

demonstrate in Sec. IV how our methodology can be applied recursively to create ultra-resilient components. Sec. V discusses the importance of protecting critical information further to protecting infrastructures, with the example of genomic information. Sec. VI highlights the role of vertical formal verification in our approach.

II. Divide-and-conquer to beat extreme threats

Targeting extreme dependability and security, we assume advanced and persistent threats including pre-deployment subversion of hard- and software components. That is, we allow adversaries to tamper with the IP blocks we use to construct our hardware platform and assume exploitable vulnerabilities in all software layers, including processor and device firmware. We assume however access to a secure fab for chip manufacturing and confine IP blocks to the digital circuit abstraction. Mixed-signal hardware designs can easily bridge isolation and allow adversaries to introspect and interfere with all hardware components, including the trusted-trustworthy hybrids on top of which our approach is built. The application areas we envisage span all systems from small, real-time controllers to large scale cloud servers.

To counteract advanced and persistent threats we follow a hybrid approach to system structuring and architecting. Researching the following divide and conquer based methodology, we 1.) start by assuming, system-wide, the existence of a few components (the 'hybrids') that are *justifiably trusted*, because they are trustworthy, whereas the rest of the systems (the 'payload' part) is not trusted. We draw from earlier research that provided the necessary architectural constructs to embody this *hybrid distributed (or modular) systems model* [8]. One fundamental rule that tells this model apart from mere trusted-third-party-based approaches is that the *trusted* components must be made *trustworthy*, by design and construction. The potential of this model 2.) has been confirmed by a number of algorithms and protocols relying on hybridisation (dividing), leveraging the power of trusted-trustworthy components to yield efficient and minimal yet secure fault and intrusion tolerant protocols [9], [10] (conquering). Recently, we have been taking the approach to new horizons of system architecting. We 3.) *decompose* payload protocols and components into smaller units to identify split points and critical parts which, once we establish their trustworthiness (e.g. by design and verification), recombine with the remaining, untrusted components such that the properties we desire emerge. Indicators for such components are: narrow interfaces, hindering attacker penetration; isolation capabilities, preventing a successful penetration of one component from spreading to others; and simplicity of the functionality, to minimize code size and hence the attack surface of the critical parts. We further complicate the task of attackers by 4.) *recursing* the problem and subjecting critical parts to the same analysis as described above. It turns out that often, initially trusted components, can be declassified to untrusted after decomposition. For example, in the next section, we see how the trusted message delivery scheme we assume, in order to reduce the number of replicas in our protocols MinBFT and CheapBFT, becomes untrusted once a trusted component is introduced (e.g., the USIG in case of MinBFT).

Key to the conquering part of our methodology is to ensure that higher-level components benefit from the underlying decomposition in such a way that desired properties 5.) *emerge* from the conglomerate of possibly replicated parts. Ideally, this emergence bypasses lower-layers as much as possible to preserve the 5.) *direct interaction* of high-level components. Direct interaction is crucial to partially regain the performance and efficiency we lose by replicating and isolating lower levels.

In the following, we exemplify this methodology in the construction of hybridisation-aware algorithms, models and architectures.

III. BFT, or Automatic Security and Dependability

Relevant challenges in BFT protocols include reducing their high resource consumption, increasing their efficiency, and reducing their considerable trusted computing base (TCB). Recent BFT protocols address these challenges using novel mechanisms. MinBFT and CheapBFT address the high resource consumption [9], [10], reducing the number of required replicas from $3f+1$ to $2f+1$. CheapBFT brings this number down to $f+1$ replicas in the normal-case, i.e., in the absence of faults, while relying on f passive but actively updated replicas to join the protocol when a replica is suspected to be faulty. This reduction is made possible through the inclusion of trusted components (e.g., in MinBFT, a message ordering ensuring component, called USIG), which must be tamperproof even if their replicas are compromised. BFT-SMaRt modularizes the BFT implementation and introduces a clear separation between the state machine and the BFT algorithm [11]. This approach differs from the traditional monolithic approach introduced with Practical BFT (PBFT) [12]. Another particularity of BFT-SMaRt is that it provides CFT and BFT: it handles crash and Byzantine faults in a seamless manner, increasing efficiency in benign situations. On the other hand, the modularisation introduced with BFT-SMaRt is a step forward in reducing the TCB associated to BFT algorithms. These smaller modules can help simplify the use of formal verification strategies to verify the security properties these algorithms guarantee. We classify this approach as *designing for verifiability*. Another benefit of a modular design is that the algorithms are prepared to take advantage of modern multi- and many-core architectures.

Currently, we are taking this approach a step forward with our work on the decomposition of MinBFT. Rather than executing the state machine and the protocol in one monolithic instance on top of the management OS, the management OS is merely used for its network stack and device driver infrastructure. The state machine and the components of the split BFT protocol are executed in enclaves, as discussed in Sec. IV. Again, *decomposition* improves security by allowing some stateless components (such as the management OS) to be rebooted if compromise is to be expected.

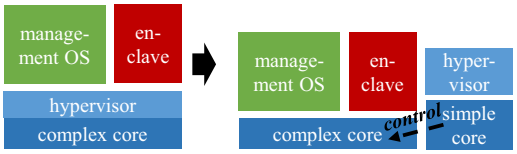

Fig. 1. Towards ultra dependable components

A crucial prerequisite for automatic (non-stop and unattended) security and dependability are countermeasures to prevent resource exhaustion. Although the above protocols are able to tolerate up to f faulty replicas, they cease to work if a persistent attacker succeeds in compromising the $f + 1^{st}$ replica. The protocols and architectures we envisage must therefore support automatic rejuvenation and diversification of replicas [13]. Rejuvenation re-instantiates a replica into a pristine state— causing attackers to lose control of the former if previously compromised— and then makes it re-join the healthy replica set. Automatic diversification is required to nullify all previous successes by the attacker and to ensure that previous attacks cannot be repeated or even performed faster. Therefore, as long as rejuvenation and diversification outpace the attacker in compromising replicas, critical infrastructures will survive.

IV. Ultra-resilient trusted components

Hybridisation and the automatic security and dependability approaches detailed in Sec. III, crucially depend on necessarily trusted and ideally tamperproof components as subversion-free anchors, as explained in Sec. II. In the following, we sketch the example of enclaves and decomposed MinBFT, showing how our methodology achieves resilience, without unduely increasing the system's power consumption or sacrificing its performance.

Intel SGX [14] already offers a first layer of protection by separating secure execution environments, called enclaves, from the operating system that is responsible for managing platform resources, including those used by enclaves. For the purpose of this paper, it is not important whether, like in SGX, enclaves are provided by a hypervisor-like implementation in microcode, or through a software hypervisor with Inktag [15], in a standard security platform such as ARM Trustzone [16]. The important property: the management OS only sees enclave state encrypted and signed to protect enclave integrity and prevent replay.

Anticipating extreme threats, we have to go one step further and investigate the subversion possibilities of both the involved hard- and software and in fact it has been shown that the isolation achieved by SGX is not perfect [17], [18].

Recursing one level down, we see that enclave security crucially depends on correct processor hardware, in particular for the management OS/enclave transitions and to execute the (software or firmware) hypervisor.

Figure 1 sketches an alternative. Rather than constraining the complex core with its SIMD accelerators, out-of-order execution, speculation, and other features required for high single-thread performance, we *decompose* the system by adding a further simple, but ultra reliable core to run the hypervisor and grant it control over the management OS/enclave transition.

In a next step, we would realize the need for enclave-to-enclave communication and, once authorized, allow *direct enclave interaction* by bypassing the encryption if communicating enclaves are scheduled back-to-back.

V. Privacy- and Integrity-preserving Decentralized Data Processing

Critical information infrastructures (CII) resilience is a necessary, but not sufficient condition for secure ICT-based societies. Data processing and storage, building on the former CII guarantees, still undergoes application-level threats that must not be ignored. Though personally identifiable information (PII) may immediately come to mind, we illustrate our view on the problems of critical data protection with a particular kind of PII, of emerging importance: biomedical data, especially genomics. Biomedical information is enduring a revolution: the collection and storage of biological material is getting systematic (tissues, fluids, etc.), both for clinical and research purposes, and digital representations of these samples are exploding in volume, especially in genomics (DNA). The latter is thanks to the advent of Next-Generation-Sequencing (NGS) machines, lowering the price and increasing the speed of sequencing, by several orders of magnitude.

This growth implies novel needs for protecting against new privacy-related attacks on genomics data, for distributed bioinformatics architectures, and for efficient privacy-preserving algorithms. Several problems and threats loom.

First, the need for economically storing and processing these huge amounts of data has put cloud computing on the forefront of scalable IT infrastructure. Second, a dramatic increase of the availability of personally identifiable information (PII) has been occurring in parallel, due to the digitalization of societal activities, the web in general, and social networks in particular, threatening anonymisation. In 2000, an alarm was raised [19], by demonstrating the re-identifiability of anonymized medical patient data. Thirteen years later, not much had changed, when Gymrek et al. [20] managed to re-identify 13.1% of the de-identified 1000-Genomes project database. Last but not least, there is a great pressure to get hold of biomedical data for reasons of different nature and coming from diverse angles, such as researchers, corporations and even governments. This confluence of interests is sometimes detrimental of harmonious solutions to the problem we actively study: the privacy vs. sharing dilemma of biomedical data.

Advanced approaches to privacy-preserving genomics data processing and architecting have been developed lately. For example, the BioBankCloud project proposes storage architectures which are based on clouds-of-clouds (i.e., multiple instances of clouds), both private and public, from several stakeholders and providers, but which are perceived seamlessly as a single cloud, by the e-biobank users and administrators [21]. In another work [22], the authors propose a privacy-

preserving method where they encrypt all genomic data before storing it, and manually mask the few genomic variations identified as sensitive. We plan to transform this prevention into tolerance by continuously re-encrypting stored data in enclaves to render partial exposure of this threshold encrypted information useless, even if the attack continues.

In a recent work, a high-throughput method was proposed to automatically segregate genomic information right after it is sequenced, at the exit of NGS machines [23]: very sensitive data is kept within the private premises of the data generating entity, whereas less sensitive data can be stored in the cloud. Genomics data can also be protected with powerful state-of-the-art encryption, coding and dispersion mechanisms, such as those proposed in [24].

The advent of high throughput NGS machines made DNA sequencing become cheaper, but also put pressure on the DNA sequencing life-cycle, which includes having millions of short DNA sequences, called reads, normally aligned to a reference genome. On the performance side, more efficient algorithms were developed, and computations parallelized on public clouds (e.g., BWA [25], Bowtie [26]). On the privacy side, since DNA data is utterly sensitive, several cryptographic mechanisms have been proposed to align reads securely (e.g., [27], [28]), but slower than the former, which in turn are not secure. We have developed a finer grained classification of the sensitivity of reads, based on a risk-analysis study, which leads to performance improvements, and to cheaper storage of DNA data, if combined with a privacy-aware storage hierarchy in the organization.

VI. High-confidence Vertical Verification

Correctness and subversion freeness of ultra-resilient components and protocols, guaranteeing the emergence of desired properties from an actively maintained majority of healthy replicas, are crucial to trust the architecture as a whole. Formal verification with proof assistants (also called theorem provers) and other sound verification tools provide the highest level of trust assurance currently known to mankind. Hence, basing our assurance argument on these technologies suggests itself.

Rich verification ecosystems have recently been developed, enabling the verification of compilers such as CompCert [29] microkernels such as seL4 [30], and even of proof assistants themselves such as Coq verified in Coq [31] and HOL verified in HOL [32].

Ideally, ultra-resilient components are implemented in high-level functional programming languages, which always have been amenable to verification, and then proven from their specification all the way down to machine level or the hardware description itself. However, the necessity to interface with specialised hardware components, e.g. low-level ultra-resilient components in cyber-physical systems, imposes specific and sometimes intricate requirements on control over data structure layout, allocation and synchonization of resources, etc. This brings the need to support and reason in terms of the typical imperative languages used in these environments, such as C and C++. VST [33] is a separation logic based verification framework that allows one to verify C programs within the Coq proof assistant and compile them to verified machine code using CompCert. Similarly, CakeML [34] is a bootstrapped verified programming language implemented in HOL, which bridges the gap between functional programming languages and also supports the verification of low-level programs [35].

New challenges arise from the verification of BFT protocols. Anticipating advanced and persistent threats, all components must be formalized as imperfect entities that eventually may be compromised by an adversary. Continued correctness must then emerge from proofs about system components that, with the exception of justifiable and temporary valid assumptions, will start exhibiting arbitrary (i.e., Byzantine) behavior.

First steps towards the verification of ultra-resilient fault-tolerant protocols have recently been made: Cryptographic primitives such as SHA, HMAC and RSA have been verified for example in Dafny as part of the Ironclad project [36] as well as in Coq using VST [37] (only SHA and HMAC). Also, Raft has been implemented and verified in Coq as part of the Verdi project [38] and Paxos has been implemented and verified in both Nuprl [39] and Dafny [40]. However, both protocols assume a crash-fault model while we have to focus on Byzantine faults.

Our current target are BFT protocols such as the evolution of the MinBFT protocol, as sketched in Section III. We strive for a C implementation on top of a hypervisor (or alternatively in SGX enclaves). The crucial insight, which allows us to cut the extreme costs that verification projects typically entail, is that decomposition allows us to leave large parts of the system unverified while focusing only on crucial components and protocol parts.

VII. Conclusions

We have presented our proposal for endorsing resilience as a unifying paradigm to address the expected increase in sophisticated targeted attacks, or advanced persistent threats (APT), against the everyday more complex critical information infrastructures. In the presence of such threats, we claim a need for dynamically and automatically handling extreme adversary power, sustaining perpetual and unattended operation in a systematic and automatic way. We have shown how this can be achieved, through a concerted approach including: hybrid and modular architectures; fault and intrusion tolerant protocols and rejuvenation and diversity mechanisms; trusted-trustworthy components; formal verification; and privacy-preserving data processing.

References

[1] R. N. Charette, *This car runs on code*, http://spectrum.ieee.org/transportation/systems/this-car-runs-on-code, Accessed: 2016-06-10.

[2] S. C. Misra and V. C. Bhavsar, "Relationships between selected software measures and latent bug-density: Guidelines for improving quality," in *Proceedings of the 2003 International Conference on Computational Science and Its Applications: PartI*, ser. ICCSA'03, Montreal, Canada: Springer-Verlag, 2003, pp. 724–732.

[3] S. H. Kan, *Metrics and Models in Software Quality Engineering*, 2nd. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002.

[4] E. N. of Excellence ReSIST, *Public deliverables*.

[5] A. Bessani, P. Sousa, M. Correia, N. F. Neves, and P. Verissimo, "The crutial way of critical infrastructure protection," *IEEE Security and Privacy, vol. 6, no. 6, pp. 44-51, Nov/Dec 2008.*, Dec. 2008.

[6] A. Fawaz, R. Berthier, W. H. Sanders, and P. Pal, "Understanding the role of automated response actions in improving ami resiliency," in *Proceedings of the NIST Cybersecurity for Cyber-Physical Systems Workshop*, Gaithersburg, Maryland, Apr. 2012, pp. 23–24.

[7] B. Donohue, *NSA director Rogers urges cyber-resiliency*, https : / / threatpost . com / nsa - director - rogers - urges - cyber - resiliency/108292/, Accessed: 2016-10-6.

[8] P. Verissimo, "Travelling through wormholes: A new look at distributed systems models," *SIGACT News*, no. 37, pp. 66–81, 2006.

[9] G. S. Veronese, M. Correia, A. N. Bessani, L. C. Lung, and P. Verissimo, "Efficient byzantine fault-tolerance," *IEEE Trans. Comput.*, vol. 62, no. 1, pp. 16–30, Jan. 2013.

[10] R. Kapitza, J. Behl, C. Cachin, T. Distler, S. Kuhnle, S. V. Mohammadi, W. Schröder-Preikschat, and K. Stengel, "Cheapbft: Resource-efficient byzantine fault tolerance," in *Proceedings of the 7th ACM European Conference on Computer Systems*, ser. EuroSys '12, Bern, Switzerland: ACM, 2012, pp. 295–308.

[11] A. Bessani, J. Sousa, and E. E. P. Alchieri, "State machine replication for the masses with bft-smart," in *Proceedings of the 2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, ser. DSN '14, Washington, DC, USA: IEEE Computer Society, 2014, pp. 355–362.

[12] M. Castro and B. Liskov, "Practical byzantine fault tolerance," in *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, ser. OSDI '99, New Orleans, Louisiana, USA: USENIX Association, 1999, pp. 173–186.

[13] P. Sousa, A. Bessani, M. Correia, N. F. Neves, and P. Verissimo, "Highly available intrusion-tolerant services with proactive-reactive recovery," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 4, pp. 452–465, Apr. 2010.

[14] V. Costan and S. Devadas, "Intel SGX explained," Massachusetts Institute of Technology, Tech. Rep., 2016, https://eprint.iacr.org/2016/086.pdf (Accessed: 2016-07-22).

[15] O. S. Hofmann, S. Kim, A. M. Dunn, M. Z. Lee, and E. Witchel, "Inktag: Secure applications on an untrusted operating system," in *ASPLOS*, 2013.

[16] T. Alves and D. Felton, "TrustZone: Integrated hardware and software security," *ARM white paper*, vol. 3, no. 4, pp. 18–24, 2004.

[17] N. Weichbrodt, A. Kurmus, P. Pietzuch, and R. Kapitza, "Asyncshock: Exploting synchronisation bugs in intel SGX enclaves," in *Proceedings of the 21st European Symposium on Research in Computer Security (ESORICS 2016)*, to appear, 2016.

[18] Y. Xu, W. Cui, and M. Peinado, "Controlled-channel attacks: Deterministic side channels for untrusted operating systems," in *IEEE Symposium on Security and Privacy*, 2015, pp. 640–656.

[19] L. Sweeney, "Simple demographics often identify people uniquely," *Health (San Francisco)*, vol. 671, pp. 1–34, 2000.

[20] M. Gymrek, A. L. McGuire, D. Golan, E. Halperin, and Y. Erlich, "Identifying personal genomes by surname inference," *Science*, vol. 339, no. 6117, pp. 321–324, 2013.

[21] A. Bessani, J. Brandt, M. Bux, V. Cogo, L. Dimitrova, J. Dowling, A. Gholami, K. Hakimzadeh, M. Hummel, M. Ismail, *et al.*, "Biobankcloud: A platform for the secure storage, sharing, and processing of large biomedical data sets," in *Workshop on Data Management and Analytics for Medicine and Healthcare*, 2015.

[22] E. Ayday, J. L. Raisaro, U. Hengartner, A. Molyneaux, and J.-P. Hubaux, "Privacy-preserving processing of raw genomic data," in *Data Privacy Management and Autonomous Spontaneous Security*, 2014, pp. 133–147.

[23] V. V. Cogo, A. Bessani, F. M. Couto, and P. Verissimo, "A high-throughput method to detect privacy-sensitive human genomic data," in *ACM Workshop on Privacy in the Electronic Society*, 2015.

[24] A. Bessani, M. Correia, B. Quaresma, F. André, and P. Sousa, "Depsky: Dependable and secure storage in a cloud-of-clouds," in *Proceedings of the Sixth Conference on Computer Systems*, 2011.

[25] H. Li and R. Durbin, "Fast and accurate short read alignment with burrows–wheeler transform," *Bioinformatics*, vol. 25, no. 14, pp. 1754–1760, 2009.

[26] B. Langmead, C. Trapnell, M. Pop, S. L. Salzberg, *et al.*, "Ultrafast and memory-efficient alignment of short dna sequences to the human genome," *Genome biol*, vol. 10, no. 3, R25, 2009.

[27] Y. Huang, D. Evans, J. Katz, and L. Malka, "Faster secure two-party computation using garbled circuits.," in *USENIX Security Symposium*, vol. 201, 2011.

[28] E. De Cristofaro, S. Faber, and G. Tsudik, "Secure genomic testing with size-and position-hiding private substring matching," in *Proc. of the 12th ACM Workshop on Privacy in the Electronic Society*, 2013, pp. 107–118.

[29] X. Leroy, "Formal verification of a realistic compiler," *Commun. ACM*, vol. 52, no. 7, pp. 107–115, 2009.

[30] G. Klein, K. Elphinstone, G. Heiser, *et al.*, "Sel4: Formal verification of an OS kernel," in *SOSP 2009*, ACM, 2009, pp. 207–220.

[31] B. Barras, "Sets in Coq, Coq in sets," *Journal of Formalized Reasoning*, vol. 3, no. 1, pp. 29–48, 2010.

[32] R. Kumar, R. Arthan, M. O. Myreen, and S. Owens, "Self-formalisation of higher-order logic - semantics, soundness, and a verified implementation," *J. Autom. Reasoning*, vol. 56, no. 3, pp. 221–259, 2016.

[33] A. W. Appel, "Verified software toolchain - (invited talk)," in *ESOP 2011*, ser. LNCS, vol. 6602, Springer, 2011, pp. 1–17.

[34] R. Kumar, M. O. Myreen, M. Norrish, and S. Owens, "Cakeml: A verified implementation of ML," in *POPL'14*, ACM, 2014, pp. 179–192.

[35] Y. K. Tan, M. O. Myreen, R. Kumar, A. C. J. Fox, S. Owens, and M. Norrish, "A new verified compiler backend for cakeml," in *ICFP 2016*, ACM, 2016, pp. 60–73.

[36] C. Hawblitzel, J. Howell, J. R. Lorch, A. Narayan, B. Parno, D. Zhang, and B. Zill, "Ironclad apps: End-to-end security via automated full-system verification," in *OSDI '14*, USENIX Association, 2014, pp. 165–181.

[37] L. Beringer, A. Petcher, K. Q. Ye, and A. W. Appel, "Verified correctness and security of openssl HMAC," in *USENIX Security 15*, USENIX Association, 2015, pp. 207–221.

[38] J. R. Wilcox, D. Woos, P. Panchekha, Z. Tatlock, X. Wang, M. D. Ernst, and T. E. Anderson, "Verdi: A framework for implementing and formally verifying distributed systems," in *PLDI 2015*, ACM, 2015, pp. 357–368.

[39] N. Schiper, V. Rahli, R. van Renesse, M. Bickford, and R. L. Constable, "Developing correctly replicated databases using formal tools," in *DSN 2014*, IEEE Computer Society, 2014, pp. 395–406.

[40] C. Hawblitzel, J. Howell, M. Kapritsos, J. R. Lorch, B. Parno, M. L. Roberts, S. T. V. Setty, and B. Zill, "Ironfleet: Proving practical distributed systems correct," in *SOSP 2015*, ACM, 2015, pp. 1–17.