Cryptography and Security in Clouds
IBM Forum Zürich
http://www.zurich.ibm.com/~cca/csc2011/program.html

J.H. Davenport — J.H.Davenport@bath.ac.uk

15–16 March 2011
v2 with updates 29 March

Abstract

1. Dalia might be interested in section 5.
2. Saif might be interested in section 12.
3. Jim/CM50209 might be interested in footnote 3 (page 5).
4. JHD/JAP/RJB should look at section 9.
5. Everyone should be amused by the jokes at footnote 8 (page 6) and 11 (page 8).

Contents

Sponsors:

1. `http://www.ecrypt.eu.org`, which produces a yearly report on Algorithms and Key Lengths;
2. Trustworthy Clouds `http://www.tclouds-project.eu`;
3. IBM Zürich.

1 Virtual Security: Data Leakage in Third-Party Clouds and VM Reset Vulnerabilities

(Invited Talk) Thomas Ristenpart (University of Wisconsin, Madison, USA)
Described by the speaker as a "bad news" talk. Comes in two parts. See [RTSS09].

1.1 Data Leakage in Third-Party Clouds

This part is based on a Fall 2008 study on Amazon's EC2. We suggested a three-phase attack:

1. cloud cartography (what is running where);
2. get malicious VM on same physical machine as victim VM;
3. side-channel attacks.

Simplified model is that users run virtual machines on cloud provider's infrastructure. Multi-tenancy is the key to the economics, and the problem from a security point of view. This has new trust models[1]: users must trust third party providers:

- not to spy themselves on users;
- be secure against other users.

Naïvely the attacker launches lots of copies of the attacking VM, but this is expensive and noisy — can the attacker do better?

We chose Amazon's EC2, with no internal knowledge. Had to abide by AUP on Amazon. We were able to do "cloud cartography" to work out where target VMs were running. Used Unix-based VMs, which are run under a Xen-based VM manager. There were three "availability zones", and five instance types (RAM and compute power). Each VM gets an external domain name and IP address, and an internal IP address, which in practice are statically assigned (8:1) to physical servers. Hence co-residence can be detected by traceroute — 1

[1] JHD query: how is this different from 1970s time-sharing?

hop means on same physical server[2]. The internal IP addresses are partitioned across the availability zones (intuitively reasonable). Running all instance types from two different accounts shows

- there is regularity in the assignment to instance types
- different accounts gets the same address ranges.

So our attack method is

- Convert external name to internal IP address using internal DNS service
- reverse-engineer the internal IP address
- therefore know which availability zone and instance type we're trying to attack (cloud cartography)
- try to place near the target taking advantage of parallel placement vulnerability: if I want a machine of type X at the same time as you (e.g. after maintenance, or scheduled refresh, or deliberate crashing of target instance), we're likely to get the same physical machines, as Amazon tries to pack VMs onto physical servers.

Could get a 40% probability of co-residence.

Extends attack of [OST06]:

1. Fill cache with attacker data;
2. Busy loop to allow victim to run;
3. Time re-reading of cache.

These techniques also allow for a co-residence check: A does HTTP GETs to B and from the cache can tell whether B is on the same machine as A.

1.2 Countermeasures by provider

1. Allow opt-out of multitenancy
2. . . . ?

JHD Should the internal DNS be more restrictive?

[2]Subsequently changed by Amazon, but there are other means.

1.3 VM Reset Vulnerabilities

By using VM snapshots, we can essentially replay the "randomness". This is not limited to clouds: internal desktop virtualisation is also vulnerable to this. Note that vendors, e.g. vmware, recommend this for 'sandboxing' browsers. [GR05] discuss how this can lead to vulnerabilities. Note that the snapshot VM might include a one-time key. They were discussing a purely hypothetical attack.

We [RY10] looked at this. Suppose first use of a VM snapshot (which already has gathered the randomness) is to go to `mybank`, then second is to `malware`. Firefox and Chrome both have this vulnerability. Note that this is a weakness in the application of TLS, and violates the specification.[3]

Should distinguish between "volume snapshots" (of persistent storage) and "full-state snapshots" (not currently in great use). If an Apache server uses full-state snapshots, then it is possible to recover its DSA secure key. HTTPS on the forked child snapshot should update the randomness with fresh data (time, PID etc.), but this is guessable. VirtualBox gave it away 10/10: vmware varies from 0/10 to 6/10 depending on exact parameters[4].

Note that snapshotting is, by definition, transparent to clients, so they don't know their randomness has been captured."Traditional cryptography fails spectacularly given bad randomness". We [BBN+09] therefore propose "hedged cryptography" (though in questions he admitted that this wasn't the whole solution, rather a part of defence in depth). Truly hardware-based RNGs (used at the appropriate point!) are also a solution.

2 A Small Latte or a PetaCycle? You Decide. The Economics of Cloud Computing and What This Means for Security

(Invited Talk) Radu Sion (Stony Brook University, USA — ex-IBM)

> I have experience only in teaching graduate students ... and as a result I realise that I don't know how to teach. [Feynman]

Speaker notes that "cloud" is basically an update of IBM's old "on demand" strategy: $\$2.5{\cdot}10^9$ expenditure.

Moore's law really talks about density of gates, but seems to apply to compute speed as well. Nielsen's law — high-end connection speed grows 50%/year (which is slightly slower[5] than Moore!). Note there is also latency to worry about.

[3] JHD observes that this is a classic case of cryptography being a system issue.

[4] JHD questioned why the range: largely dependent on clock jitter.

[5] In after-dinner discussions, the speaker and JHD agreed that the failire of most people to realise that the ratio of two exponential growths is itself an exponential growth/decline is worrying.

Note that there are all sorts of outsourcing. What has changed since 1995 Sun's "the network is the computer"? Cheaper networks, better virtualisation etc. What is a "cycle" — pretty variable, but will normalise to Amazon EC2.

2.1 Costs

Technology costs, security costs. Technology costs are storage, compute and networking. Claims[6] Google went to Intel and asked for CPUs which tolerate 1C hotter (hence 3% cooling cost saving). Claims Amazon gets a far better deal (factor of 10) from Verizon for networking costs than anyone else. Claims the following costs of computing (measured in p¢/cycle).

Home 7.11 (many costs are discounted)

Small business 26 (staff is 50% of the total cost)

Large business 7 (JHD's reading of his barchart is that all costs shrink roughly in proportion between small and large business)

Cloud 0.58, down to 0.3 at `rackspace` (staff is less than 10% of the total).

But the per-bit costs of data transfer are three orders of magnitude greater (800–6000p¢/bit). Furthermore data storage is 6p¢/bit/year, and most of this is energy (so Google can't really undercut[7]). A disc seek is 270n¢.

Definition 1 (First principle of cloud viability) *It is not worth outsourcing any task of less than 4000 cycles per 32-bit word transfer.*

This applies when I am the user.

But if I have external users (e.g. at the application level I am the server in a client/server scenario), I have to pay for them to communicate to me. Cloud providers get cheaper networking. Therefore in practice cloud deployment saves 4500p¢ per client/application transfer bit, and 10p¢ per CPU cycle.

2.2 Benefits

Availability, opportunity, consolidation.

Is the cloud more or less secure — yes!

Tells a story about the manufacturer of voting machines who published photographs of the actual key used to lock the machine on the website.[8]

The big discussion on clouds in Wall Street is always about liability.

[6] Apparently googling "Google hotter chips" is the clue. See `http://www.datacenterknowledge.com/archives/2008/10/15/did-google-intel-discuss-hotter-chip-specs/`.

[7] There was a debate about savings from better RAID technology, but not much conclusion.

[8] Presumably `http://www.bradblog.com/?p=4066`.

> While it is possible in principle for computation to be done on encrypted date [...] current techniques would more than undo the economy gained by outsourcing and show little sign of becoming practical. — Diffie

3 The Cloud was Tipsy and Ate my Files!

(Invited Talk) Giuseppe Ateniese (Johns Hopkins University, USA & La Sapienza) [ABC+07] How to prove possession of data. "PDP" – provable data possession.

> Your data are stored in the Cloud. My company, for a fee, monitors the Cloud to ensure that the entire contents of your digtal life is intact and readily available [Elevator Pitch]

What's cool about this is that there is no knowledge of the data being stored.

Users with limited resources or expertise can outsource their storage, with universal access independent of location. This comes with full data backup/recovery. This may help with compliance with electronic records storage legislation. Outsourcing avoids initial setup costs and provides scalability. Case study — Library of Congress. Only small parts of the data are ever retrieved, and data are stored for a long time (forever).

Third parties might move data off disc to tape, discard data that is rarely accessed, hide data loss incidents. Also possibly deliberate modification. Can I use my cellphone to verify that the Library of Congress is complete?

Trivial schemes that do not work,

- Download everything back (needs a local copy)
- Ask the storage server to compute the MAC of the archive. The cost of this computation is infeasible.
- Ask the storage server to send the data and the MAC of random file blocks

Our solution is to aggregate the MACs. We force the server to do certain computations on the on the file blocks, and verify that these computations are correct via authentication tags. We call this "homomorpic authentication tags", and claim this introduces "public verifiability" (our notion).

Recall RSA 101: $N = pq$ with $p = 2p' + 1$, $q = 2q' + 1$ and $ed \equiv 1 \pmod{\phi(N)}$. Send i to Google. $t_i = (H(W_i) \cdot g^{m_i})^d$, and get back (m_i, t_i) and check that $0 \leq m_i < N$, and $\frac{t_i^e}{H(W_i)} = g^{m_i}$. This is the naïve way.

More sophisticated, select some blocks at random, with a random integer a_i each. Then ask Google for $T = \prod t_i^{a_i}$, and $M = \sum a_i m_i$. Verify $\frac{T^e}{\prod H(W_i)^{a_i}} = g^M$. In practice we would send a "recipe" for the a_i, e.g. a key of a PRNG, or of a HMAC. This method allows for public verifiability (all that is needed is the public key of the scheme), an unbounded number of challenges, and there are no private data.

This is doable on a cellphone, but requires crypto operations. There is a variant using symmetric-keys, but this throws away public verifiability, and the number of challenges is bounded. This essentially works by precomputing the answers. This can be done via homomorphic sigma protocols.

It is also the case that this scheme can be adapted to allow for dynamic operations: block update, deletion etc.

4 Writing on Wind and Water: Storage Security in the Cloud

(Invited Talk) Ari Juels (RSA, The Security Division of EMC, USA)
(Etymology is from Catullus *Carmina* LXX 3): lovers' vows should be written on wind and water.[9]

The cloud has a nice simple API. But a cryptographers's view of life is more varied, including trojans, maliciious providers. RSA Labs Research program is "Remote Cloud Security Checkups". The idea is to restore transparency to cloud to achieve stronger security/compliance, without trusting the cloud itself.

Case study. Alice has her data stored on `megarchive`, but they outsource to `miniarchive`, who uses customers' storage (Bob), who uses peer–peer file storage, and it ends upon Alice's computer.

We want POR — proof of retrievability (possibly in compliance with an SLA over time). Periodic dowloads are one way, but resource-intensive. Better is spot-checking: save a few blocks, and randomly check one of these. Still uses storage, and doesn't detect small corruptions. Better is to compute checksums of the blocks, rather than the blocks themselves. Still doesn't detect small erasures.

Combine this with ECC to detect small errors. Note that speed of response to the challenge tells one something about the *quality* of retrievability.

But this doesn't tell us whether the server *might* have a problem in the future. Hence we might use RAID-type technology, striping across cloud providers. But how do we tell that they are genuinely independent, and how do we deal witha mobile adversary? RAID[10]-type redundancy allows us to tolerate t provider failures. POR will let us bound t by "whack-a-mole" techniques[11]. We call this HAIL — High Availability Intermediate Layer. Amazon charges 15¢/GB/month, but Memopal offers 2¢/GB/month on a lower service level. Would HAIL on Memopal (+clones) be cheaper than Amazon?

Another challenge is the physical layer. Amazon claims to store three distinct copies for resilience. How can we check this? Neither POR nor downloading will

[9] Sir Richard Burton's translation is "womanly words that are spoken to desireful lover Ought to be written on wind or upon water that runs", and apparently dates from Catullus' disenchanted phase. The original is "sed mulier cupido quod dicit amanti, in vento et rapida scribere oportet aqua."

[10] EMC pioneered RAID, and now own RSA.

[11] JHD first heard "guacamole", but "whack-a-mole" is what the author intended. Apparently it's a reference to a video game.

prove this. Note that virtualisation complicates questions of independence. So let property X be "a tenant's file can survive a certain number of drive crashes". The speaker explains his solution by an analogy based on duplicate pizza ovens! [JK07]

[Questions] POR differs from PDP in including ECC, and also there is a technical difference in that a POR actually checks the interface. Updates to files are hard in POR, but somewhat easier in HAIL.

5 Using TPMs to Tame Uncertainty in the Cloud

(Invited Talk) Rodrigo Rodrigues (Max Planck Institute for Software Systems, Germany)
"Gaining Customer Trust with Excalibur" was the revised title.

Problem: customers surrender full control of their data. Can data leak? Will it stay within jurisdictional boundaries? Which software will access it?. BC: the application developer retained all control. AC: Cloud provider retains all control. The cloud provider publishes a list of node configurations, the customer chooses this, and the provider enforces this. Attrubute-value pairs for cloud nodes, e.g. software,location, hardware. Customers define policies over configurations.

5.1 Trust Model

Software administrator → adversarial rôle. Platform developer → mitigation rôle. Cloud provider → trusted to deploy defenses (but no control over all adminstrators). Threats are integrity and confidentiality.

A TPM is a trusted platform in a configuration that implements expected behaviour (integrity, confidentiality etc.). Enforces this despite adversarial system administrator behaviour.

"Policy-sealed Data" is the key abstraction. Data must be sealed before they are uploaded, and when leaving a node that obeys the policy (migration; suspend-to-disc). The data must be unsealed to access it. We will use TPMs to implement this idea. TPMs are widely available, and enable remote query about node characteristics. These are already used, e.g. BitLocker. Recall that TPM has a strong identity, via a per-node identity key. There is also a "trusted boot". Based on this, we have remote attestation, allowing a remote challenger to verify the authenticity of the platform. TPM also has the concept of "secure storage".

Based on attribute-based encryption (ABE). A Master Key + attributes → decryption key. The goal is therefore to run sheathe/unsheathe locally, without intervention of the master key owner. The problem is that this is slow in terms of generation of keys, and the process itself. (slower than RSA). To prevent a node authentication bottleneck, we pre-generate the decryption keys. We encrypt a symmetric key which is used to encrypt the actual data.

We have a VM-rental service analogous to EC2, based on Eucalyptus/Xen, providing location restrictions. This involved 52 lines of code in create, save, restore and migrate. Another feature that could be added is preventing curious sysadmins from accessing the VM image. For a 10MB image, the encryption overhead is 35%. For VM-rental, the encryption is about 1.5 times as big as the native Xen cost, but nearly all of this was in the symmetric encrption (unavoidable if we're doing anythng like this), rather than the ABE itself.

Q. Why ABE?

A. The key point of ABE is that data are sealed to the policy, not the specific node.

Q. Revocation?

A. We don't, because of the TPM. Software upgrades do take some managing.

Q-JHD What crypto technology?

A Need to get back to you!

6 Towards Multi-Layer Autonomic Isolation of Cloud Computing and Networking Resources

Aurélien Wailly, Marc Lacoste, Hervé Debar (Orange Labs and Télécom Sud Paris, France)

There are processors, devices, memory, protocol layers and physical network equipment. These define a "computing view" and a "network view". We want a way to guarantee resource isolation. There are components at the VM Layer, Hypervisor Layer and Physical layer, which may be networking or computing components.

We have autonomic security: detection/decision/reaction feedback lop. The benefits ae automatic adaptationof security mechanisms and easier adminstration, but loop composition is a difficulty. For example, if the physical layer detects an attack and isolates the VM, but the VM also wants to update the antivirus database. Therefore we have horizontal orchestration (HO) to synchronise across views, and vertical (VO) to synchronise across layers. In fact VO synchronises the HOs. This is where the administrator-oriented isolation policies are defined.

As well as user VMs (UVM), we have administrative VM (AVM), which synchronises HO and VO. Hence an anti-virus alter from the UVM will go to HO-VM, which would tell VO, which might then instruct HO-Hypervisor to isolate that VM.

7 Security Considerations for Virtual Platform Provisioning (in Telecommunication Networks

Mudassar Aslam, Christian Gehrmann (Swedish Institute of Computer Science)
Virtual Platform provisioning is common in data centres, and now in clouds, but telecom networks is never (at least in this terminology). What should be virtualize, what are the scenarios, and the business models? there might be 2G nodes, 3G nodes and an LTE (?) network. [ZLGTG10]. perators don't like paying for underutilised service, wheras physical providers have a long lead-time. If this was virtualised, both sides might be happier.

The participating entities are telecom equipment provides, software providers, operators, end-user and (possibly) virtualised entity providers.

Presents a scenario, with various attacks and corresponding security requirements.

Conclusion: moving telecoms into clouds needs security issues to be addressed, but this can't be done without integrating the security models with the business models.

8 Mobile Trusted Virtual Domains

(Invited Talk) Ahmad-Reza Sadeghi (TU Darmstadt, Germany)[12]
This talk is about putting existing technology together to solve a real problem. The "holy grail" of cryptography is secure multiparty communication without Trusted Third Parties. What we want is secure communication between TVD (Trusted Virtual Domains). This is a coalition of virtual and/or physical machines, and trust is based on a security policy beyond physical boundaries. This is more abstract that typical access control mechanisms. In practice, technologies such as VPN will be deployed in order to deploy these. This will, in practice, rely on a trusted computing base of some sort. Each TVD will need a (possibly distributed) master. There are such services as

- membership and revocation
- communication
- storage.

If a platformwishes to join TVD, it has to load a copy of the TVD master to configure the machine to conform to the TVD requirements, and enters a 'join' protocol when it is satisfied.

What about mobility? He discussed a "secure kernel", which supports Linux, Symbian and Android, and allows very fast switching between different compartments and TVDs. He gave some Android basics, and said that what they do on to ofAndroidis they colour the applications according to TVD membership.

[12] Did a poll in the audience. About 50:50 between Android and iPhone, with **no** Windows. speaker: "I told them so".

However, the 'native code' option of Android lets the hypervisor be bypassed, and the reference monitor is also too coarse-grained (existing attacks). They therefore propose isolation via containers (details missing).

Current work includes

- "trust but verify" allowing a trusted execution environment,
- fine-grained policy enforcements
- migration and attestation of VMs
- efficiency
- evaluating security provided by current clouds (thi sis being done at the moment, and is apparently yielding some surprising results)

* but most importantly

- usability
- legal issues.

9 Technical Challenges of Forensic Investigations in Cloud Computing Environments

Dominik Birk (Ruhr-Universität Bochum, Germany)
There are various issues; "Safe harbour Agreement" versus Patriot Act? There aremalicious events and accidental events. Cloud Service Providers (CSPs) artificially make the cloud opaque, for commercial confidentiality and other reasons. The traditional SAP (Securing, Analyzing, Presenting) Cycle, working off a bit-wise copy, is not possible any more. There has traditionally been much emphasis on the "chain of custody" requirement. Note that a real private cloud does support forensic investigation. Note that the CSP will have log data that is not accessible to you, particularly in an SaaS context. Note that in IaaS, one should think that the user could install forensic tools, but, at least in EC2, if the machine shuts down, the data is lost (this has happened in real life).

Summary: lack of transparency, loss of evidence data, compliance issues, secure data deletion, SLA verification, Missing Best Practices. Note that the Amazon SLA (and others) offer no help. Hence cloud forensics is basically impossible. If you don't trust the CSP, then leave the cloud!

Q-IBM(US) We have to educate the lawyers, even in the non-cloud situation.

All, *sotto voce* Easier siad than doe.

10 Self-Managed Services Conceptual Model in Trustworthy Clouds' Infrastructure

Imad M. Abbadi (Oxford University Computing Laboratory, UK); part of Tcloud project
In the beginning, a sysadmin had self-written scripts to manage storage and systems/applications.In the modern, butpre-cloud, world, the sysadmin is now using tools to manage system architecture. this more-or-less survived the transition to virtualisation.

So why should we move to automated management services.

MTTD Mean Time to Discovery

MTTI Mean Time to Investigate

MTTR Mean Time to Resolve.

We need to consider the heterogeneous nature and complex layeringof clouds. Do standards help/ are they deployed? What about cloud insiders (including sub-contractors etc.).

Q-JHD They are standards for whether things **work**, but not whether they **work efficiently**.

A Agreed.

11 Predicate Encryption for Private and Searchable Remote Storage

(Invited Talk) Giuseppe Persiano (Università di Salerno, Italy)
I'm not questioning *why* the cloud — it is there, e.g. Gmail, DropBox. Rather, how to make it better.

Example 1 *Simple case — a* data owner *storing his data on* untrusted storage. *Assume*

- *DO assumes US doesn't destroy data (if necessary use duplication etc.)*
- *DO assumes US doesn't modify data (ECC, MACs etc.)*
- *US doesn't read the data*[13] *(e.g. straightforward encryption).*

Consider a table of the conference organisers (first name,last name, affiliation). Store it with each cell encrypted. Query is "who were the iBM organisers"

- *Download whole table and decrypt. Needs bandwidth and local storage.*

[13] Recounts a story where, when reading e-mail on Gmail, he got adverts about the city being discussed in the e-mail.

- *Tell US to decrypt — spoils the point of encryption.*
- *this is why we want predicate encryption.*

Predicate encription P: key K with attribute y can decrypt ciphertext C with attribute x iff $P(x, y) = 1$).

Example 2 *Alice generates MSK, and public key PK', which she publishes. Bob, her boyfriend, wants to send M, so sends $E(PK', M, \texttt{private})$. Dean, a colleague, wants to send M', so sends $E(PK', M', \texttt{work})$. Alice uses MSK to create decryption keys for* `private` *and* `work`*, and gives the latter to her secretary.*

Example 3 (1 continued) *Store an extra column $E(PK', (a, b, c), 0)$ with a predicate such that $P((a, b, c), (d, e, f)) := c = f$. Then I send a key corresponding to $(*, *, IBM)$ and then only the IBM rows* can *be decrypted (as far as this key is concerned), and then are returned (still under normal encryption).*

In practice, this needs "hidden vector encryption". Standard challenge game to explain security. [BW07] gave a construction based on pairing with composite order groups. [IP08] had a better construction. n-bit prime p, random $t_{i,b}, v_{i,b} \in \mathbf{Z}_p$ Our implementation uses PBC (`http://crypto.stanford.edu/pbc`) for basic pairing and elliptic cuvres, and jPBC (apparently can be basically a wrapper). We used $y^2 = x^3 + x$ over $\mathbf{F}_q$ for $p \equiv 3 \pmod 4$ (512 bits). Primes were taken off appropriate standard.

Used a basic Macintosh. With 100 attributes, jPBC-PBC (not clear whether this included pre-computation) takes 200ms to generate a search key, 500 with jPBC, and 2500 with jPBC without pre-computation. Other items in ratio. Initial encryption is expensive, but then this should be rare.

11.1 More Theory

[DCIP11] Selective security assumes that the adversary attaks the data rather than the key. The adversary declares that he wants to disstinguish (*,*.IBM) from (*,*,SAL). Full security allows the adversary to base his attack on the public key (chosen independently of the data) and on the keys obtained.

Note also that US knows all the searches I have made. Claims that key security is impossible for a public-key-based solution. In fact, the the DO/US scenario, private key is OK. Example 2 needs public key, though. hence a hybrid approach. Two further questions.

- Is sublinear search possible? If not, we won't excite the database people!
- A lazy DO might just return"no". Is there any verifiability?

12 Side Channels in Cloud Services: The Case of Deduplication in Cloud Storage

(Invited Talk) Benny Pinkas (Bar Ilan University, Israel)
See **??**.
Many cloud backup services. Mozy has 1M personal cusomers, with 75PB storage. Dropbox has over 3M customers. Shows example of Mozy saying "I already have this file" — once correctly, and once (apparently) not. This deduplication can be applied at file or block level. Claimed to save 90% of storage in common business scenarios. This is claimed to be the "most impactful storage technology" —EMC bought DataDomain for $2.1B. We assume

- cross-user deduplication (common, indeed necessary to get the savings)
- client-side deduplication (saves bandwidth).

Naïvely, these attacks will only work once, since the file is now uploaded. But we can interrupt the upload if an actual upload was triggered.

1. Can check if a file has been uploaded by some-one else. State Department could have done this over Wikileaks.
2. Can search for files. Alice can try uploading all of "Dear Bob, your bonus is $0", "Dear Bob, your bonus is $500" etc., and see which has been uploaded.
3. Covert channel. If Alice has malware on Bob's machine, which can't transfer data directly to Alice, it can signal via Mozy/Dropbox.

Note that the upload costs, on Amazon, about as much as two months storage. hence server-side deduplication is too expensive. However, Mozy is now deduplicating small files (< 1MB) at the server, after we informed them of our attacks. We have a better solution now, which is randomized threshold server upload. It is necesaary that the threshold is per-file random, and secret. It is also possible (but may complicate the user interface) to allow users to specify that no deduplication should occur.

12.1 Hash values and Proofs of Ownership

A short hash value serves as proof of file ownership. Several attacks are possible (note that the IBM co-authors could not participate in this phase).

1. suppose the dedup procedure uses a common hash function (e.g. SHA-256). Suppose Bob writes daily lab reports, and publishes their hash as a time stamp. Alice then tries to upload a file with this hash value (note this involves hacking the protocol at Alice's client side), then gets registered as an owner. Once this happens, Alice can then download the file, which was meant to be secret.

2. Efficient file leakage. Even if bob doesn't publish the hash values, malware running on Bob's machine can leak the has values (note the covert channel via dropbox above).

3. Essentially using the dropbox as a content distribution network. Alice uploads, then e-mails the hash to friends. This may well render the server liable for aiding piracy, and will wreck the economic model anyway (many more downoads than expected).

Possible solutions are SHA(service——file) rather than SHA(file). This doesn't solve attacks 2, 3. SHA(nonce——file) causes great retrievel load at the server. Any true solution must be

- bandwidth efficient
- must not cause much retrieval of files from secondary storage. Only a few bytes needed per file.
- Files must not be stored in main memory
- Attacked might have partial (e.g. 95% in the bonus case) knowledge of the file.

Need "proof of ownership" (Pow). The server preprocesses the file, and stored a small Pow.

12.2 First solution

Construct a Merkle tree of the file. The server stores the root of the tree. The server asks the client to present the path to l random leaves. The server verifies that these agree with its stored root. If the client knows a fraction p of the leaves, the probability of success is p^l hence this doesn't solve the "client knows most of the file" problem.

12.3 Variant

Apply an erasure code to the file, and then construct a Merkle tree over the encoding. Let's say that our encoding is such that knowing 50% is sufficient to recover the file. Then this makes the probability of failed guessing to be 2^{-L}. The problem with this is that erasure coding tends to require pre-processing the wholefile. Therefore we will perform this on a hashed subset of the file, say 64MB. This has been implemented. For 2GB file, this adds 50% to the existing read from disk then SHA256 solution. saves a factor of 6 when compared with uploading over a dedicated 100Mb/sec link.

12.4 Conclusions

1. Deduplication is a very powerful technqiue, but does have various leaks.

2. Most vendors are not aware of the leaking.

3. The challenge is to offer meaningful privacy at reasonable cost.

Q This doesn't solve the bonus leak question.

A But the "server side dedup (therefore always upload) small files" solution does.

Q In the CDN setup, what's to stop Bob bouncing the questions to Alice?

A Good point. The server could do timng analysis, but it's not clear this is reliable. So effectively we're assuming Alice is off-line.

13 Byzantine Fault Tolerance for the Cloud

Hans P. Reiser (University of Lisboa, Portugal and University of Passau, Germany)

There are lots of problems: dozens of new vulnerabilities every week, 1.5M node botnet detected in 2005, vulnerabilities een in critical infrastructures: look at Stuxnet. Clearly prevention is best, but in practice we need fault tolerance as well. BFT is about f out of n components failing in unpredictable (malicious) ways, but nevertheless the system working well. [Lamport1980] for introoduction, [CastroLiskov1999] for first pratcuical system. But this doesnt really get deployed (yet). Castro's NFS server alowed $n = 3f + 1$ servers to tolerate f failures, and had performance close to unreplicated NFS servers. Oceanstore, Farsite, Rosebud use BFT to provide a P2P storage servive. `UpRight` is a library that aims to make BFT simple and viable for real services. This is used to provide BFT versions of:

- Zookeeper lock service;
- Hadoop File Sever (HFS).

13.1 Trust Model

Basic BFT classifiers servers as good/bad (up to f items). There are hybrid models (A2M, EBAWA) which assume a small subsystem is never bad (e.g. a TPM). These are typically much more efficient.

The cloud is more complicated. Do we assume the provider might be bad? Do we assume that the infrastrcuture might have imperfections. What about the applications running on the cloud? Even my own applications *might* have imperfections. We will trust the provider (i.e. a hybrid model).

A further problem is recovery. If there are f faults, what happens at fault $f + 1$? Maybe we can benefit from the cloud model and dynamically allocate new instances to deal with faulty ones (needs trust in the replica generation!). But most BFT work is ststic.

We have an ongoing `CloudFIT` project. This architecture has a "wormhole" in the hypervisor, and a BFT library that runs in the VM. This handles group-membership, reconfiguration etc. We use the EBAWA algorithm with $2f + 1$ nodes and a trusted USIG component.

14 Integrity and Consistency for Untrusted Services

Christian Cachin (IBM Research - Zurich, Switzerland)
25 years ago we knew where our data were: now we don't. Note that LA wants to move to Google,but LAPD is not ready becaus eof security concerns. Solutions are baically separate coulds, e.g. "google Federal".

Ours sytem model has a server S which is normally correct, but sometimes faulty. There are clients $C_1, \ldots, C_n$ which are correct, but may crash, may disconnect, and have small trusted model. In terms of CIA, we are only addressing Integrity here. Example: outsourcing of audit records.

Our storage model is a memory array MEM $=\{x_1, \ldots, x_n\}$. There are two (atomic) operations :`read(i)` and `write(i,x)`. Clients may digitally sign their write requests, Hence the server can't forge read values, but can replay old versions, and send different values to different clients.

The sever may present different views to clients (e.g. servers "fork" tehir history). Clients cannot prevent this..

The key idea is "fork linearizability" [MS02]. If the sever forks the views of two clients, then the views a re forked ever after and they will never see each other's updates. This means that every inconsistency results in a fork, which can never be "covered up".

We say that

safe every read not concurrent with a write returns the most recently written value.

regular safe and and read concurrent with a write returns one or another.

A trivial protocol ([MS02]): the complete history is signed. The sever sends the history with allsignatures andthe client verifies all operations and signatures, and adds its operation and signs the new hstory Unfortunately impractical since the hsotory, and messages, grow with system age. Instead the client C_i stores a timestamp t_i and a version is a vector of timestamps.Then a client must check that its own timestamp is correct and the whole vector must be greater than its last one. If forking happnes, vectors become incomparable. Timestamps are now $O(n)$.

there is [CV09] a CSVN version of SVN using this protocol. There is also work on moving from storage to an arbitrary service. We need to work on extending the protocol to avoid "blocking".

15 Verifiable Computation with Two or More Clouds

Ran Canetti, Ben Riva, Guy Rothblum (Tel Aviv University, Israel and Princeton University, USA)
"Cloud copputing the timesharing of today". We assume powerful clouds and weak clients. The basic concern is verfifiable computations. Can a ewak client check isf the powerful cloud is cheating? So the could produces output, then engages in an interactive proof gae with the weak client, and the client then accepts/rejects the cloud's output. [Kilian1992,Gennaroetal2010,Applebaumetal2010] One canalso try to push the trsut deeper into trusted agrdware.

Practitioners want something useful, whereas theoreticians want unconditional proofs, fewer rounds etc.

So change the model to interacting with several clouds. A rival approach here is majority voting. We show

- unconditional sound ness for L-uniform NC
- a protocol for polynomial-time computation, sound assuming CRH, with $\log n$ rounds
- a runing implementatuion of this model on x86, with a 7× overhead.

[FeigeKilian1997] They show a one-round refereed game for PSPACE, and $poly(n)$ for PTIME.

15.1 log-time reduction for two clouds

Turing machine model (reality: x86): state×head position×tape contents. The example [FeigeKilian1997] is binary search. Once we have found that step k is correct and $k+1$ is the first point at which the clouds disagree, we can run that one step. But how big can a configuration be? Now each TM transition uses $O(1)$ information. We therefore compute a Merkle hhas tree(MHT) Once Bob has sent Alice the root of the hash tree, and query can be returning t_i and the hash values along the path to the root. Hnece the proof of consistency is $O(\log n)$. In fact, we use (state,head position,root(MHT(tape))) as somewhat easier.

Our prototype QUIN changes state/head position to be CPU regsiters, tape becomes stack and heap memory, and TM transition becomes an instruction. In practice we only send the MHT for the final configuration as well. For a naïve[14] determinant caculaton, the overhead factor goes from 188 (size 10) to 7(size 12) Our main tool , Quin Executor, has its own large overheads.

[14] JHD wonders if this means $n!$. Apparently it does.

16 Twin Clouds: An Architecture for Secure Cloud Computing

Sven Bugiel, Stefan Nürnberger, Ahmad Sadeghi, Thomas Schneider (TU Darmstadt and Ruhr-Universität Bochum, Germany)
Picture showed a large dark cloud and a smaller white cloud, which is symbolic. Radu showed that cloud providers get more tcoks for the buck, especially with multi-tenancy. Note that Amazon chanrges $1.60/hour to rent about $5K worth of equipment. Our aim is to minimise resposne time while protecing ourselves against malicious providers. Note the importanc eof data protection in E-Health (records, genome data), E-Goverenment, E-Business (e-processing even for conventional business, such as payroll). We want Confidentiality, Integrity, Verifiability and Efficiency. Building the cloud from secure hardware is basically infeasible. What are the alternatives? [Yao1986] [Gennaroetal2011] based on [Gentry2009]. This does not work for multiple clients.

16.1 Token-Based Cloud Computing

Extend the cloud with a secure hardware token, fully trusted by the client. Thismight be a shared (virtualised) cryptographic coprocessor, or even a smart card. We will therefore assume that this is in a "trusted cloud'[15], which is fully trusted by the client, secure, slow and expensive: to be contrasted with "commodity cloud", which is untrusted but powerful and cheap.

So the client sends program P and data D to the trusted cloud. This computes $\hat{D}$ etc., and has a multi-phase transaction with the commodity cloud. This is the setup phase. For processing a query, q goes in clear to the trusted cloud, which does symmetric-key encryption to transform the query and the response.

Q What's the point?

A the setup phase is indeed expensive, but the per-query cost is linear in in/out size, and low latency.

17 Secure Outsourced Computation in a Multi-Tenant Cloud

Seny Kamara, Mariana Raykova (Microsoft Research and Columbia University, USA)
The client sends (x, f) and gets $f(x)$. Virtualisation enables multi-tenancy, which allows cost savings. hence we need multi-tenancy, but it introduces security concenrs. Current solution is VM isolation. But the hypervisor can be

- exploited
- bypassed (see [Ristenpartetal2009]).

[15] In practice, might be own machine/private cloud.

FHE and VC, but FHE is not practical, and VC is based on FHE. These area lso overkill, since interaction is OK, and the cloud is not a single-server environment. So let's assume a delegation protocol. C provides an input. The Cloud has Vms $V_1, \ldots, V_m$ whch cary out a protocol amongst them selves,and return a result. Assumptiions.

- Cross VM attacks always work (pessimistic, but reasonable), which may be *semi-honest* (read-only), or *malicious*, i.e. involve total subvesion of the machine.

We use the ideal/real world paradigm from [Canetti2001]. We want to show that, if the adversary A locates at most $w-1$ VMs with ours, then A learns no information and C receives correct output.

Our work is based on Multi-Party Computation, which we claim is now very relevant. We also need 'secret sharing' [Shamir].

1. Split x into shares $(S_1, \ldots, s_n)$
2. Store each share in a separate VM
3. Make the VMs evaluate F using MPC: $y := F(s-1, \ldots, s_n, r_1 \oplus \cdots \oplus r_n)$.
4. Use the r_i to geenerate shares w_i in y
5. Return all the w_i to C, which reconstructs y, and then recovers the true answer.

Note this is only useful if the cost of all this is less that the cost of renting an unshared VM. There is overhead for the sharing and recovery, and for the extra $n-1$ VMs being used. Ongoing work is on efficient protocols for specific cases, e.g. polynomials.

Q Where does this work in practice?

A It almost certainly needs special-case delegation protocols.

Q But how do you ensure that these VMs are placed on separate physical machines?

A Good question!

18 Amortized Sublinear Secure Multi Party Computation

Dov Gordon, Jonathan Katz, Vladimir Kolesnikov, Tal Malkin, Mariana Raykova, Yevgeniy Vahlis (Columbia University, University of Maryland, and Bell Labs)
Secure Computation: Bob and Alice want to compute F, depends on inputs which should be kept secret. In particular, Alice wants to search in data Bob possesses. This is possible for any function [Yao1986] [BGW1988]. Howver, this assumes F is a Boolean circuit. This is not very efficient, and this seems ot be inherent.

- Privacty shouldrequire us to touch all parts of the database, since otherwsie we are leaking information.
- The whole database is essentially encoded in the (inputs to the) circuit, and hence $\Omega(n)$.
- Equally, weneed to touch every brnach of the program.

Hence we seem to need $\Omega(n)$ work. Can we do better via amortised complexity over multiple executions? We us ethe RAM computation model ratehr than circuits. This allowsruning time to be sublinear (clearly a good idea!), and does not need to access every branch. However, we are now leaking information via runtime.

18.1 Our Contributions

- A generic approact to sublinear 2-party computations
* A compiler for ecure computation of RAM protocols based in any oblivious RAM.
- Optimised effciiency construction
* two-party oblivious RAM propotocl ith access pattern privacy from both parties[PR2010]
* Yao computation for circuits of minimised size

RAM model data stored in a single array V, and an instruction is read/write (d, v), and select next instruction. One example is binary search.

There has been much work on Oblivious RAm pciteDMN2011 [GO2000]. The operations are `ORAMInit` and `ORAMEval` An instruictions is a series of resad/write. Correctness is the right value beng returned, whereas oblivious means that $(A_1, \ldots, A_n)$ is indistinguishable from another sequence $(B_1, \ldots, B_n)$ of the same number fo instructions. Then each GenNextInstr and doInstr are done via 2-paryt MPC.

Theorem 1 (ORAM) *If a computation can be done in t steps on a RAM, then it can be done in $O(t \log^2 tn + \cdots)$ steps on an ORAM with ....*

We want efficient 2party encryption and decryption (note that [Yao1986] used XOR). We may virtual addresses to real ones, hiding the mapping from the client. A paper (gate-counting) exercise shows substantial savings (factor 10^3) on [Yao1986].

Q There are a lot of hidden constants in thsi gate-counting.

A Yes

19 Computation on Randomized Data

Florian Kerschbaum and Kiayias Aggelos (SAP Research Karlsruhe, Germany and University of Athens, Greece)
Example: supply chain masterplanning. Warehouses, distribution centres etc. There is institutionallevel and lcoal-level planning. This is in tehory a straightforward linear programming problem, *but* people don't want to reveal the business-sensitive data involved in true global minimisation.

We found a (textbook) problem instance with 282 variables (real-life might be 4M!). Note that simplex[16] is $O(n^3)$! Solving ths takes 4 seconds. [Toft2009] estimated actual running time to be 7 years [CatrinadeHoogh2010] brought it down to 20 days. We can achieve 25 minutes using multiparty randomness. Let Q be a positive matrix with positive inverse (diagonal times permutation) Consider $PAQy \leq Pb + PAQr$ as the new problem to be outsourced to the cloud.

Abstraction. Given g, find f and f' such that $f(g(f(x,r)),r) = g(x)$ with f being significantly cheaper than g (else there's no point!). Note that the traditional model $V^{real} = V^{ideal} + P(1/k)$ for polynomial P becomes $V^{real} = V^{ideal} \times P(1/k)$, so knowing P is very important.

There is some work on LP random transformd, [BBR2009]. the foloowing research questions arise.

- Can we derive f from g, as in FairPlay [BNP08]?
- Can we automatically assess the security of F?

Q Leakage isn't uniform, e.g. signsare probably more important than lower-order bits.

A There are technical results about this which can make leakage uniform, independent of the input.

Q Suppose the "spy" is only after specific information.

A Same answer.

Q Numerical stability?

A No theoretical study, but in practice this hasn't occurred.

20 Private and Perennial Distributed Computation

Shlomi Dolev, Juan Garay, Niv Gilboa, Vladimir Kolesnikov (Ben-Gurion University, Israel, AT&T Labs Research, USA, and Bell Labs, USA)

[16] Average running time. But the worst-case is exponential, and traditional SMC is *always* worst-case.

We have a dealer, and k parties. The dealer has an automaton A with initial state S, and the dealer wants to outsource the computation of $A(S)[inputs]$ to the parties. There is no communication between the parties during the computation, and at the end what they return to the dealer should let th dealer recover $T = A(S)[inputs]$.

There is also an advesrary model, who knows A, but not S or the input tape. This is not within the MPC paradigm, since the input is unbounded, we want information-theoretic security etc.

Our scheme allows *perennial* computation for any FSA. The complexity depends on the Krohn–Rhodes decomposition of FSA. This is linear in certain interesting cases, but $n!$ in the worst case. The complexiy measures are size of FSA and number of transitions. It turns out to be trivial for a permutation automaton. KR decomposition expresses as a composition of permutation and reset automata. Note the cascade product of automata. [Eilenberg1976].

The dealer decomposes A into a product of permutation and reset automata, does secret-sharing of '1'. The online phase consists of each party processing the input. Ends with reconstruction.

21 Discussion: Cryptographic Secure Computation for Cloud Computing?

Moderation: Matthias Schunter (IBM Research — Zurich, Switzerland)

BP Benny Pinkas – Bar Ilan

FK Florian Kerschbaum — SAP

MR Mariana Raykova — Columbia

AJ Ari Juels — RSA

The moderator proposed these questions.

1. Can multi-party computation and crypto be useful in practice (when cost is the focus)

BP Not in the cloud, but DARPA has identified requirements for such sharing among intelligence agencies. Here cost is *not* the issue.

MR There are other issues than cost, such as availability.

FK The benefits or performing thec omputation may be greater than the cost of the computation. But the benefits have to be *huge*, hence he doesn't expect this to happen in the near future. The real challenge is security against the service provider.

AJ Probably least enthusiastic og the panel. People want to keep their data in the cloud, irrespective of security. Inter-cloud this might be required, but anyone who trusts *one* cloud may well be prepared to trust a cloud and some brokers. He quoted FHE as a neat answer to a non-problem. What we really want to do is compute across the data of multiple tenants. Then there's a key management issue. We (RSA in-house) then showed that this is impossible.

FK Voting is a good example where the (non-financial) benefits may make it worthwhile.

Floor Even if this doesn't work in the cloud, there may be other applications.

Floor Google (gmail) makes money out of seeing my data (and therefore posting relevant advertisements), so why should they do more computations on the data in order *not* to be able to see it?

Floor My company[17] runs servers all round the world, because most jurisdictions do not allow us to process data abroad. Hence secure computing would be a major saving in terms of consolidation for us.

2. What are the genuinely new questions (the moderator had spotted a lot of recycling of old research questions)?

BP We are seeing more quantification of costs in terms of $ etc., but the questions are really the same. We saw outsourcing of computation back in the days of set top boxes/smart cards, so less is new than we think.

MR We should be getting closer to quantifying real costs.

FK I think security against the service provider *is* a new question. This is somewhat different from mobile agents. Whether it will be *important* is a different question.

AJ There is a Berkeley tech report "what'snew in cloud security". There was only one paper in the "genuinely new" bucket (Ristenpart). Inter-tenant protection is actually the new question.

Ristenpart There's been 20–30 years of MPC literature, but what about "engineering".

BP But engineering is difficult to get support for. Constant factors don't merit publication!

MR The DARPA calls do specify precise targets.

Ristenpart A better benchmarking set would be very helpful.

floor Equivalent to the RSA challenge.

[17] Financial organisation, PB of data. Later admitted to be UBS.

FK SAP has a secure computing facility: the problem isn;t the engineering: it's the marketing.

3. Questions frothe floor.

IBM Current cloud contracts are all "buyer beware". Unless this changes, there will be no demand for secure computing. To get there, we need a dialogue between the techies and the lawyers.[18]

22 Miscellaneous Gleanings

22.1 Basel II/Europe

Apparently a consequence of the way Basel II is implemented in Europe is that there are rules imposing minimal distances between primary and backup data centres. This has interesting interaction with the speed of light and "synchronous write" requirements. Apparently run-length encoding on **unencrypted** data solves this problem in practice, but this has its own issues.

[18] Disbelief all round.

Abstracts of Workshop on Cryptography and Security in Clouds (March 15-16, 2011, Zurich)

Virtual Security: Data Leakage in Third-Party Clouds and VM Reset Vulnerabilities

Thomas Ristenpart (University of Wisconsin, Madison, USA)

In this talk we'll cover new security issues that arise in the use of virtualization. First we'll look at third-party cloud computing services such as Amazon's EC2 and Microsoft Azure. We'll see how so-called placement vulnerabilities allow an attacker to arrange for a malicious virtual machine (VM) to be assigned to the same physical server as a target victim's VM. From there, the attacking VM can mount side channel attacks. We'll report on initial work on cache-based side channels that can measure the victim's computational load to, for example, infer the kinds of web traffic received by a web server running on the victim's VM.

Next I'll present recent work on showing a new class of vulnerabilities, termed VM reset vulnerabilities, that arise due to reuse of VM snapshots. A snapshot is the saved state of a VM, which can include caches, memory, persistent storage, etc. A reset vulnerability occurs when resuming two or more times from the same VM snapshot exposes security bugs. I'll report on our discovery of several reset vulnerabilities in modern browsers used within commonly-used VM managers. These vulnerabilities exploit weaknesses in cryptographic protocols when confronted with reused randomness. I'll then explore potential solutions.

This talk will cover joint work with Stevan Savage, Hovav Shacham, Eran Tromer, and Scott Yilek

A Small Latte or a PetaCycle? You Decide. The Economics of Cloud Computing and What This Means for Security

Radu Sion (Stony Brook University, USA)

In this talk we explore the economics of technology outsourcing in general and cloud computing in particular. We identify cost trade-offs and postulate the key principles of outsourcing that define when cloud deployment is appropriate and why. We also briefly touch on several main cyber-security aspects that impact the appeal of clouds. We outline and investigate some of the main research challenges on optimizing for these trade-offs. If you come to this talk you are also very likely to find out exactly how many US dollars you need to spend to break your favorite cipher or send one of your bits over the network.

The Cloud was tipsy and ate my files!

Giuseppe Ateniese (Johns Hopkins University, USA)

Our entire digital life will be stored on remote storage servers such as Amazon S3, Microsoft Azure, Google, MobileMe, etc. Our emails, pictures, calendars, documents, music/video playlists, and generic files will be readily available, anytime and anywhere.

In this talk, we will answer the following question: How can we check whether the entire content of our digital life is actually intact and accessible, even though we have no local copy of it?

Writing on Wind and Water: Storage Security in the Cloud

Ari Juels (RSA, The Security Division of EMC, USA)

The Cloud abstracts away infrastructural complexity for the benefit of tenants. But to tenants' detriment, it can also abstract away essential security information. In this talk, I'll discuss several protocols that remotely test cloud storage integrity and robustness without a reliance on detailed infrastructural knowledge or trust in cloud providers. Tenants or auditors can execute these protocols to: (1) Verify the integrity of files without downloading them; (2) Distribute files across cloud providers and verify their intactness with periodic, inexpensive checks (in a cloud analog to RAID); and (3) Test whether files are resilient to drive crashes.

Joint work with Kevin Bowers, Marten van Dijk, Burt Kaliski, Alina Oprea, and Ron Rivest.

Using TPMs to Tame Uncertainty in the Cloud

Rodrigo Rodrigues (Max Planck Institute for Software Systems, Germany)

Despite the benefits of cloud computing, its users face a significant downside: they must yield control of their data to the cloud provider, and therefore need to blindly trust it to correctly manage a large, complex infrastructure prone to issues such as accidental or intentional data loss or disclosure. In this talk I will present Excalibur, a system that enables cloud providers to build services that give customers more assurances regarding the set of nodes allowed to manipulate their data, e.g., by restricting the software configuration they are allowed to run. Excalibur leverages commodity trusted computing hardware (TPMs) to provide the abstraction of policy-sealed data, where data is encrypted in a way that is associated with a given policy, and can only be retrieved by cloud nodes that obey that policy. In implementing this abstraction, Excalibur addresses several challenges that arise from using TPMs in the context of cloud computing.

This is joint work with Nuno Santos, Krishna Gummadi (MPI-SWS) and Stefan Saroiu (MSR).

Towards Multi-Layer Autonomic Isolation of Cloud Computing and Networking Resources

Aurélien Wailly, Marc Lacoste, Hervé Debar (Orange Labs and Télécom Sud Paris, France)

Despite its many foreseen benefits, the main barrier to adoption of cloud computing remains security. Vulnerabilities introduced by virtualization of computing resources, and unclear effectiveness of traditional security architectures in fully virtualized networks raise many security challenges. The most critical issue remains resource sharing in a multi-tenant environment, which creates new attack vectors. The question is thus how to guarantee strong resource isolation, both on the computing and networking side. System and network complexity make manual security maintenance impossible by human administrators. Computing and networking isolation over virtualized environments should thus be achieved and automated.

Unfortunately, current solutions fail to achieve that goal: hugely fragmented, they tackle the problem only from one side and at a given layer, thus without end-to-end guarantees. Moreover, they remain difficult to administer. A new integrated and more flexible approach is therefore needed.

This paper describes a unified autonomic management framework for IaaS resource isolation, at different layers, and from both computing and networking perspectives. A nested architecture is proposed to orchestrate multiple autonomic security loops, both over views and layers, resulting in very flexible self-managed cloud resource isolation. A first design for the corresponding framework is also specified for a simple IaaS infrastructure.

Security Considerations for Virtual Platform Provisioning

Mudassar Aslam, Christian Gehrmann (Swedish Institute of Computer Science)

The concept of virtualization is not new but leveraging virtualization in different modes and at different layers has revolutionized its usage scenarios. Virtualization can be applied at application layer to create sandbox environment, operating system layer to virtualize shared system resources (e.g. memory, CPU), at platform level or in any other useful possible hybrid scheme. When virtualization is applied at platform level, the resulting virtualized platform can run multiple virtual machines as if they were physically separated real machines. Provisioning virtualized platforms in this way is often also referred to as Platform as a Service (PaaS) in the cloud computing terminology. Different business models, like datacenters or telecommunication providers and operators, can get business benefits by using platform virtualization due to the possibility of increased resource utilization and reduced upfront infrastructure setup expenditures. This opportunity comes together with new security issues. An organization that runs services in form of virtual machine images on an offered platform needs security guarantees. In short, it wants evidence that the platforms it utilizes are trustworthy and that sensitive information is protected. Even if this sounds natural and straight forward, few attempts have been made to analyze in details what these expectations means from a security technology perspective in a realistic deployment scenario. In this paper we present a telecommunication virtualized platform provisioning scenario with two major stakeholders, the operator who utilizes virtualized telecommunication platform resources and the service provider, who offers such resources to operators. We make threats analysis for this scenario and derive major security requirements from the different stakeholders' perspectives. Through investigating a particular virtual machine provisioning use case, we take the first steps towards a better understanding of the major security obstacles with respect to platform service offerings. The last couple of years we have seen

increased activities around security for clouds regarding different usage and business models. We contribute to this important area through a thorough security analysis of a concrete deployment scenario. Finally, we use the security requirements derived through the analysis to make a comparison with contemporary related research and to identify future research challenges in the area.

Mobile Trusted Virtual Domains

Ahmad Sadeghi (TU Darmstadt, Germany)

TBA

Technical Challenges of Forensic Investigations in Cloud Computing Environments

Dominik Birk (Ruhr-Universität Bochum, Germany)

Cloud Computing is arguably one of the most discussed information technology topics in recent times. It presents many promising technological and economical opportunities. However, many customers remain reluctant to move their business IT infrastructure completely to "the Cloud". One of the main concerns of customers is Cloud security and the threat of the unknown. Cloud Service Providers (CSP) encourage this perception by not letting their customers see what is behind their "virtual curtain". A seldomly discussed, but in this regard highly relevant open issue is the ability to perform digital investigations. This continues to fuel insecurity on the sides of both providers and customers. In Cloud Forensics, the lack of physical access to servers constitutes a completely new and disruptive challenge for investigators. Due to the decentralized nature of data processing in the Cloud, traditional approaches to evidence collection and recovery are no longer practical. This paper focuses on the technical aspects of digital forensics in distributed Cloud environments. We contribute by assessing whether it is possible for the customer of Cloud Computing services to perform a traditional digital investigation from a technical standpoint. Furthermore we discuss possible new methodologies helping customers to perform such investigations and discuss future issues.

Self-Managed Services Conceptual Model in Trustworthy Clouds' Infrastructure

Imad M. Abbadi (Oxford University Computing Laboratory, UK)

Current clouds infrastructure do not provide the full potential of automated self-managed services. Cloud infrastructure management are supported by clouds' internal employees and contractors (e.g. enterprise architects, system and security administrators). Such manual management process that require human intervention is not adequate considering the cloud promising future as an Internet scale critical infrastructure. This paper is concerned about exploring and analyzing automated self-managed services for cloud's virtual resources. We propose a conceptual model of self-managed services interdependencies and identify static and dynamic factors affecting their automated actions in the context of cloud computing. Next, we identify the challenges involved in providing secure and reliable self-managed services. We have just started the work in this area as part of EU funded Trusted cloud (TCloud) project.

Predicate Encryption for Private and Searchable Remote Storage

Giuseppe Persiano (Università di Salerno, Italy)

In this talk we will survey the state of the art in Predicate Encryption with special focus on Hidden Vector Encryption schemes and show its applicability to Private and Searchable Remote Storage. Our thesis is that Predicate Encryption offers solid Cryptographic foundations for Remote Storage but several issues remain to be addressed before we can deploy usable and private remote storage.

Side Channels in Cloud Services: The Case of Deduplication in Cloud Storage

Benny Pinkas (Bar Ilan University, Israel)

The talk will discuss deduplication, a form of compression in which duplicate copies of files are replaced by links to a single copy. Deduplication is known to reduce the space and bandwidth requirements of Cloud storage services by more than 90%, and is most effective when applied across multiple users.

We study the privacy implications of cross-user deduplication. We demonstrate how deduplication can be used as a side channel which reveals information about the contents of files of other users, as a covert channel by which malicious software can communicate with its control center, or as a method to retrieve files about which you have only partial information.

Due to the high savings offered by cross-user deduplication, cloud storage providers are unlikely to stop using this technology. We therefore propose mechanisms that enable cross-user deduplication while ensuring meaningful privacy guarantees.

Byzantine Fault Tolerance for the Cloud

Hans P. Reiser (University of Lisboa, Portugal and University of Passau, Germany)

CloudFIT is an ongoing project that designs an architecture for intrusiontolerant applications that can be deployed dynamically in the cloud. This position paper presents an outline of the architecture that is being developed in the project, and discusses the implications of the deployment in the cloud. We explore to what extent existing BFT algorithms can be used for increasing security and availability in the proposed architecture and what issues still need to be resolved in the future.

Integrity and Consistency for Untrusted Services

Christian Cachin (IBM Research - Zurich, Switzerland)

A group of mutually trusting clients outsources an arbitrary computation service to a remote provider, which they do not fully trust and that may be subject to attacks. The clients do not communicate with each other and would like to verify the integrity of the stored data, the correctness of the remote computation process, and the consistency of the provider's responses.

We present a novel protocol that guarantees atomic operations to all clients when the provider is correct and fork-linearizable semantics when it is faulty; this means that all clients which

observe each other's operations are consistent, in the sense that their own operations, plus those operations whose effects they see, have occurred atomically in same sequence. This protocol generalizes previous approaches that provided such guarantees only for outsourced storage services.

Verifiable Computation with Two or More Clouds

Ran Canetti, Ben Riva, Guy Rothblum (Tel Aviv University, Israel and Princeton University, USA)

The current move to Cloud Computing raises the need for verifiable delegation of computations, where a weak client delegates his computation to a powerful cloud, while maintaining the ability to verify that the result is correct. Although there are prior solutions to this problem, none of them is yet both general and practical for real-world use.

We propose to extend the model as follows. Instead of using one cloud, the client uses two or more different clouds to perform his computation. The client can verify the correct result of the computation, as long as at least one of the clouds is honest. We believe that such extension suits the world of cloud computing where cloud providers have incentives not to collude, and the client is free to use any set of clouds he wants.

Our results are twofold. First, we show two protocols in this model:

1. A computationally sound verifiable computation for any efficiently computable function, with logarithmically many rounds, based on any collision-resistant hash family.

2. A 1-round (2-messages) unconditionally sound verifiable computation for any function computable in log-space uniform N C.

Second, we show that our first protocol works for essentially any sequential program, and we present an implementation of the protocol, called QUIN, for Windows executables. We describe its architecture and experiment with several parameters on live clouds.

Twin Clouds: An Architecture for Secure Cloud Computing

Sven Bugiel, Stefan Nürnberger, Ahmad Sadeghi, Thomas Schneider (TU Darmstadt and Ruhr-Universität Bochum, Germany)

Cloud computing promises a more cost effective enabling technology to outsource storage and computations. Existing approaches for secure outsourcing of data and arbitrary computations are either based on a single tamper-proof hardware, or based on recently proposed fully homomorphic encryption. The hardware based solutions are not scaleable, and fully homomorphic encryption is currently only of theoretical interest and very inefficient.

In this paper we propose an architecture for secure outsourcing of data and arbitrary computations to an untrusted commodity cloud. In our approach, the user communicates with a trusted cloud (either a private cloud or built from multiple secure hardware modules) which encrypts and verifies the data stored and operations performed in the untrusted commodity cloud. We split the computations such that the trusted cloud is mostly used for security-critical operations in the less time-critical setup phase, whereas queries to the outsourced data are processed in parallel by the fast commodity cloud on encrypted data.

Secure Outsourced Computation in a Multi-tenant Cloud

Seny Kamara, Mariana Raykova (Microsoft Research and Columbia University, USA)

We present a general-purpose protocol that enables a client to delegate the computation of any function to a cluster of n machines in such a way that no adversary that corrupts at most $n-1$ machines can recover any information about the client's input or output. The protocol makes black-box use of multi-party computation (MPC) and secret sharing and inherits the security properties of the underlying MPC protocol (i.e., passive vs. adaptive security and security in the presence of a semi-honest vs. malicious adversary).

Using this protocol, a client can securely delegate any computation to a multi-tenant cloud so long as the adversary is not co-located on at least one machine in the cloud. Alternatively, a client can use our protocol to securely delegate its computation to multiple multi-tenant clouds so long as the adversary is not co-located on at least one machine in one of the clouds.

Amortized Sublinear Secure Multi Party Computation

Dov Gordon, Jonathan Katz, Vladimir Kolesnikov, Tal Malkin, Mariana Raykova, Yevgeniy Vahlis (Columbia University, University of Maryland, and Bell Labs)

We study the problem of secure two-party and multi-party computation in a setting where some of the participating parties hold very large inputs. Such settings increasingly appear when participants wish to securely query a database server, a typical situation in cloud related applications. Classic results in secure computation require work that grows linearly with the size of the input, while insecure versions of the same computation might require access to only a small number of database entries.

We present new secure MPC protocols that, in an amortized analysis, have only polylogarithmic overhead when compared with the work done in an insecure computation of the functionality. Our first protocol is generically constructed from any Oblivious RAM scheme and any secure computation protocol. The second protocol is optimized for secure two-party computation, and is based directly on basic cryptographic primitive

Computation on Randomized Data

Florian Kerschbaum and Kiayias Aggelos (SAP Research Karlsruhe, Germany and University of Athens, Greece)

Cryptographic tools, such as secure computation or homomorphic encryption, are very computationally expensive. This makes their use for confidentiality protection of client's data against an untrusted service provider uneconomical in most applications of cloud computing. In this paper we present techniques for randomizing data using light-weight operations and then securely outsourcing the computation to a server. We discuss how to formally assess the security of our approach and present linear programming as a case study.

Private and Perennial Distributed Computation

Shlomi Dolev, Juan Garay, Niv Gilboa, Vladimir Kolesnikov (Ben-Gurion University, Israel, AT&T Labs Research, USA, and Bell Labs, USA)

In this paper we consider the problem of n agents (servers) wishing to perform a given computation on behalf of a user, on common inputs and in a privacy preserving manner, in the sense that even if the entire memory contents of some of them are exposed, no information is revealed about the state of the computation, and where there is no a priori bound on the number of inputs. The problem has received ample attention recently in several domains, including cloud computing as well as swarm computing and Unmanned Aerial Vehicles (UAV) that collaborate in a common mission, and schemes have been proposed that achieve this notion of privacy for arbitrary computations, at the expense of one round of communication per input among the n agents.

In this work we show how to avoid communication altogether during the course of the computation, with the trade-off of computing a smaller class of functions, namely, those carried out by finite-state automata. Our scheme, which is based on a novel combination of secret-sharing techniques and the Krohn-Rhodes decomposition of finite state automata, achieves the above goal in an information-theoretically secure manner, and, furthermore, does not require randomness during its execution.

References

[ABC+07] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song. Provable data possession at untrusted stores. In *Proceedings 14th ACM conference on Computer and communications security*, pages 598–609, 2007.

[BBN+09] M. Bellare, Z. Brakerski, M. Naor, T. Ristenpart, G. Segev, H. Shacham, and S. Yilek. Hedged Public-Key Encryption: How to Protect against Bad Randomness. In *Proceedings ASIACRYPT 2009*, pages 232–249, 2009.

[BW07] D. Boneh and B. Waters. Conjunctive, Subset, and Range Queries on Encrypted Data. In S.P. Vadhan, editor, *Proceedings Theory of Cryptography '07*, pages 535–554, 2007.

[DCIP11] A. De Caro, V. Iovino, and G. Persiano. Hidden Vector Encryption Fully Secure Against Unrestricted Queries: No Question Left Unanswered. `http://libeccio.dia.unisa.it/Papers/FullySecureHVE/unrestricted.pdf`, 2011.

[GR05] T. Garfinkel and M. Rosenblum. When virtual is harder than real: security challenges in virtual machine based computing environments. In *Proceedings HOTOS'05*, 2005.

[IP08] V. Iovino and G. Persiano. Hidden-Vector Encryption with Groups of Prime Order. In S.D. Galbraith and K.G. Paterson, editors, *Proceedings Pairing 2008*, pages 75–88, 2008.

[JK07] A. Juels and B.S. Kaliski Jr. Pors: proofs of retrievability for large files. In *Proceedings 14th ACM conference on Computer and communications security*, pages 584–597, 2007.

[OST06] D.A. Osvik, A. Shamir, and E. Tromer. Cache Attacks and Countermeasures: The Case of AES. *Topics in Cryptology CT-RSA 2006*, pages 1–20, 2006.

[RTSS09] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage. Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In *Proceedings 16th ACM conference on Computer and communications security - CCS '09*, pages 199–212, 2009.

[RY10] T. Ristenpart and S. Yilek. When Good Randomness Goes Bad: Virtual Machine Reset Vulnerabilities and Hedging Deployed Cryptography. In *Proceedings ISOC NDSS 2010*, 2010.

[ZLGTG10] Y. Zaki, Z. Liang, C. Goerg, and A. Timm-Giel. LTE wireless virtualization and spectrum management. *Wireless and Mobile Networking Conference (WMNC)*, pages 1–6, 2010.