Securing Database as a Service

Issues and Compromises

Database as a service has several major issues and concerns, such as data security, trust, expectations, regulations, and performance issues. Proposed solutions include risk management, better contractual agreements, database encryption, and authenticity techniques.

Joel Weis and Jim Alves-Foss
University of Idaho

Today, cloud computing generates a lot of hype: it's both promising and scary. Businesses see its potential but also have many concerns. Although the concept of time-shared remote services isn't new, cloud computing infrastructures use new technologies and services, some of which haven't been fully evaluated with respect to security. Database as a service (DaaS) is a prime example of a service that's both exciting and full of difficult security issues. This article considers cloud computing's benefits and shortcomings and mentions specific solutions to help it take hold as a valuable technology.

The Cloud Rationale

Economics is the primary cause for interest in cloud computing or software as a service (SaaS)—the first idea in a long time with the potential to change IT forever.[1] It promises a cheap, reliable, and flexible alternative to today's collections of heterogeneous, rigid, monolithic, and expensive-to-maintain systems.[2] Each current corporate IT system resulted from hardware purchases, software development or procurement, and funding for the professionals that maintain it.[3] In a cloud system, the service provider supplies these things, and the service is extremely flexible. It can automatically balance loads or be more scalable and is accessible anywhere with Internet access.

Businesses, government agencies, and other users have the same reason for using cloud computing—cost. It's an obvious benefit for small business startups in which a large investment is necessary to set up a new IT infrastructure. Even enterprise-class businesses find the advantages of cloud computing alluring. For example, using cloud services, businesses can eliminate the need for the professionals who maintain and support the underlying complexities for some of the most desirable new IT technologies, such as highly scalable, variably provisioned systems. An obvious benefit to customers is that computing resources, such as virtual servers, data storage, and network capabilities, are all load balancing and automatically expandable. Resources are allocated as needed, and loads can be transferred automatically to better locations, producing a robust, reliable service.[4]

The Cloud Paradigm

Cloud computing can incorporate a variety of modern Internet technologies, such as grid computing, Web applications, and virtual machines. It's simply the next natural step in Internet. Networking and the Internet began many years ago to further computing and collaboration. As technology has matured, so has our use of computers for teamwork and cooperation. Now we're approaching a new age in which the Internet is used as a medium to provide computing as a service to customers.

The idea is simple enough. A large company with many computing resources, such as large datacenters, reaches an agreement with customers. Customers can run their programs, store data, host virtual machines, and so on, using the provider's resources. Customers can terminate their contract, avoid startup and main-

 1540-7993/11/$26.00 © 2011 IEEE COPUBLISHED BY THE IEEE COMPUTER AND RELIABILITY SOCIETIES

tenance costs, and benefit from the provider's ability to dynamically allocate their resources.

Types of Service Provider Models

Cloud computing as a buzzword was derived in part from the idea of the telecommunications "clouds" that VPNs were depicted as passing through when they first came into use. As a result, there's still no clear definition for this term. The same goes for most related terms. Thus, we'll provide a brief explanation of some of the terms in this article.

One classification for cloud types is *public* or *private*.[4] Public clouds exist when a third party, such as a utility, offers computing resources as a service. In a private cloud, a sole user owns and operates the computing resources—a business uses its own resources to create many of the same advantages and technologies of a public cloud. Because of their limited applications, usually only enterprise-class businesses with their own datacenters consider private clouds.

Cloud computing service providers typically offer one or more of the following resource types, which create a more common classification of cloud types: infrastructure, platform, or software. *Infrastructure as a service* (IaaS) provides a basic computing environment, allowing customers to build, load, and run their own virtual machines (often servers) in the cloud. The provider ensures that the machines have the necessary computing power and access to Internet bandwidth. *Platform as a service* (PaaS) provides an application environment. The provider supplies development tools and ensures the means to run and maintain the programs in a Web environment. This paradigm has also been called Web 2.0. *Software as a service* provides customers with a particular piece of software. The provider runs the software and provides Internet access to it, but the customer feeds it data and instructions.

Where Does Database-as-a-Service Fall?

DaaS is a prime example of SaaS.[5] The service provider picks the database management software and installs, runs, and manages it. Although this description might make it sound like the database is some generic COTS database management system (DBMS), this usually isn't the case. Often, the DBMSs are custom applications the service provider has developed. Private clouds have used this type of service for extremely large, nonrelational databases, such as search engines. In any case, it's still just a software application provided to the customer. For economic reasons, DaaS is particularly well suited to many small- to medium-sized businesses that rely on databases but find their installation and maintenance costs restrictive. Maintaining a database requires trained professionals, so the service is even more valuable because customers don't have to hire, train, and pay them.

It's important to remember that the goal of DaaS is to make things easier. Relational databases have been in heavy use for decades because they provide constraints and control and simplify the management of large amounts of related data. However, several major aspects of this cloud model can actually make things more difficult. Encryption is the primary method for preserving confidentiality and integrity and in aiding data security, but it's also extremely computationally expensive for the service provider. Performing relational database tasks (running a query language) on encrypted data requires more complex algorithms. Much research has focused on methods to perform queries on encrypted data as well as creating encrypted data indexes.[1,6–10] Currently, the only way to maintain a fully encrypted database that can still function with most of the important features is to make a performance sacrifice that leaves the database unusable. The bottom line is that it usually makes data management more difficult.

Another critical part of the model is the database itself, which the service provider uses and maintains for the customer. The typical (relational) database can be expensive to maintain and isn't very scalable.[2] Surely a service provider with many customers would have trouble maintaining and operating all of its databases. The current solution is a pseudodatabase—a simple generic-container management system with little functionality.[11] It's low maintenance and highly scalable, but the standard SQL-like query languages must be replaced.[2] The lack of built-in constraints, controls, and relationships means that businesses will have little motivation to ditch their current local relational databases for a cloud solution. Recent research has focused on scalable relational databases; however, security is absent in these models. Microsoft SQL Azure is an example of an early adoption of such a model.

In the cloud market, dozens of companies offer secure storage, and many have added functionality to the storage and called it database services. This is common because this setup doesn't face the same issues as a scalable, secure, relational database run in the cloud. Nor is it as attractive. In fact, few cloud concepts face as many problems, which is why we focus on DaaS; if it's usable, other cloud services should be as well.

Obstacles in the Cloud

Given the benefits, it might be surprising that cloud computing adoption has been very slow. Businesses are reluctant to use cloud services. One glaring reason is that they're unsure their data will be secure. But

this is one of many concerns.[4,12] Many businesses have been storing private data in Web email for years now, which is arguably a form of cloud computing.[4] An interesting thing about many of these concerns is that they're problems faced (and accepted) elsewhere in computing—they just appear in a new light in cloud computing. We can mitigate some of these concerns; others are risks that we must manage.

Adoption of a New Technology

Businesses fear the cloud for many of the same reasons that accompany any new technology. New technologies usually must cross a "hump" of acceptance before becoming widespread. The primary concern of most businesses is unstable operation.[4] Customers worry about hiccups with these massive new cloud systems, and indeed there are. Several notable examples include some of the largest providers, such as older[12] and more recent Amazon outages. Customers also worry that their service provider won't endure. Although the cloud computing market is expanding, we can't judge provider stability this early on.

Another reason is fear of earning a bad reputation by being associated with someone else using the same computing resources. It's not good business to be associated with lawbreakers, and in cloud computing, it's impossible to know who's sharing the same resources. In March 2009, law enforcement shut down a provider to investigate the possible criminal activity of one of its customers. Several of the provider's other customers went out of business because of the investigation and the lengthy downtime.[4]

Service Provider Expectations

Most businesses interested in shifting to cloud services are used to having certain features available. This becomes problematic when a cloud provider handles services. A provider might not offer some services, or they might be difficult to implement. In either case, a lack of familiar features isn't helping cloud computing catch on.

From our viewpoint, the number-one missing feature is data security. There are two issues here. One is preventing others (such as other customers) from reading private data. This is a clear and obvious concern that's prominent in scenarios such as theft or other direct malicious attacks. The other issue concerns the service provider reading private data. In addition to the customer not trusting the provider itself, the provider isn't immune to attacks or other malicious activity, targeted or otherwise. Both issues apply to other security concerns and are commensurate with the confidentiality level desired. In most cases, some level of encryption is a given. In DaaS, encryption can affect performance dramatically, as we discuss later. Extreme security includes guaranteeing the physical security of computing locations, residual data on disks when the service provider migrates data or replaces hardware, and even memory and processor management in cloud machines. Another key facet that customers demand is completeness. It's important not only that customers have data integrity assurance but also that they receive the entire collection of any data they request.

One downside to demanding extensive confidentiality and security from a cloud provider is that it tends to hinder investigative actions. Audits and investigations require as much knowledge and detail about the data and data events as possible. However, the strict security customers desire in the cloud prevents the creation or availability of this data. Businesses often want some kind of investigative and auditing option, just in case, but because of cloud computing's nature, this is almost impossible. The same goes for knowledge of unauthorized access. Customers want the service provider to notify them if such breaches occur, but this kind of monitoring and notification system typically isn't available. Even if it were, if the service provider is considered untrusted—which should be the default case—there's no reason to trust its monitoring solution.[5]

Legal Issues

Legal concerns shouldn't be taken lightly because they're the driving force behind many security issues. Businesses already face regulatory requirements, privacy laws, and data security laws, and government agencies face even more. Probably the biggest dilemma is who (the customer or the service provider) is responsible for personal data. We must also consider the Health Insurance Portability and Accountability Act (HIPAA); the Health Information Technology for Economic and Clinical Health Act (HITECH); the Gramm–Leach–Bliley Act (GLB); the Federal Information Security Management Act (FISMA); federal, state, and local laws; and so on.[13] Exactly how each of these applies depends on the situation, the service-level agreement (SLA) between the customer and provider, and data location. The latter brings a new concern because users don't control and often don't know where their data is physically located. This introduces a slew of legal problems, especially if their data crosses international boundaries.[13]

An interesting thing about many of these concerns is that they're problems faced (and accepted) elsewhere in computing—they just appear in a new light in cloud computing.

Living with DaaS Insecurity

It's clear that cloud computing has a long fight ahead to see widespread adoption. Luckily, many key battles are similar to ones from other areas of IT that have already been fought and won. Cloud computing still faces all the problems traditional IT faces but in a new context. In time, the application of regulations and laws will be tested and decided, and providers' services—such as recovery and backup options—will expand to suit the common customer demands. For some applications, cloud computing might never be suitable unless it's private. Here, the customer can trust and verify the service provider because the customer is operating it. Examples might include multilevel secure (MLS) databases or high-assurance applications concerned with information leaks via multicore processors or memory sharing. Although researchers have shown it's possible to set up a system that mitigates these threats appropriately, it's unlikely that a public service provider system will ever be adequately verified.

Several obstacles that have interfered with cloud computing adoption just need a larger perspective. Although many relate to data security in the cloud, we shouldn't forget that all cloud data must pass through the Internet and its various security technologies just to get to the customer. Researchers have studied Internet data transmission security, and the truth is that we can only expect a certain level of security. It's full of weaknesses and flaws and can offer only a minimal amount of assurance.[14] We must take this into account when demanding a higher level of security for data storage that we can only access via the Internet. Thus, we must learn to restrict our cloud security expectations.

The Service-Level Agreement

The SLA is the only legal document between the service provider and customer, which makes it a key aspect of the service. It's the best way to mitigate and manage risks, understand the assurance level available, and discover and deal with the insecurities. Of course, the document differs among providers and types of cloud services. The lack of SLA standards is cause for concern.[15] Today's SLAs read much like an agreement for a minor utility, focusing on billing, costs, waivers, and exceptions, with only a limited discussion of service-level expectations.

Because it's the only legal agreement that customers have with the service provider, customers should leverage it to get the best service. Ultimately, customers should have clear expectations of the service provider and the service, and a clear understanding of the risks and assumptions being made while using the service.

Customers should know what the service provider is doing—or will do—about[15]

- disaster recovery,
- physical security,
- logical security,
- privileged user access,
- restrictions on data location,
- segregation of customer data,
- customer auditing,
- auditing of the service provider,
- support for investigations,
- regulatory compliance,
- data destruction,
- encryption key management, and
- network security.

Simple business issues should also be addressed, including a precise definition of services offered and an agreement on how their performance is measured and reported. The SLA should clearly outline both parties' responsibilities and include a system for managing any problems between them. Other business items, such as warranties, remedies, company acquisitions, and termination conditions, should also be discussed.

Some issues require an agreement between the customer and provider, whereas others require proof of services from the provider. Some, such as key management, require customers to provide information to the service provider. Customers need to understand the level of service being provided and the risks involved, including the strength of the security solutions being offered. A service provider won't provide perfect security, but with an understanding of what's provided, customers can make informed decisions.

Database-Specific Issues

Most DaaS-specific obstacles must be overcome via a compromise. First, to maintain acceptable performance, some sacrifices must be made. There's a variety of reasons why such sacrifices are necessary, but the greatest reason is security maintenance. To keep data secure and usable, certain database features that customers are accustomed to might be absent. Indeed, nearly all the obstacles that DaaS faces can be overcome by making some sort of compromise on performance or features. This adds to customers' reluctance to adopt the technology. In fact, some areas of DaaS haven't yet been studied because of the reluctance to make seemingly severe compromises—for example, updating the database in the cloud.[1,5]

Security Solutions for DaaS

A service provider has certain options for implementing the database and its security.

Encryption

Encrypted databases are nothing new. Various encryption technologies have existed for more than a decade. With a pile of data sitting in storage, encryption seems like the perfect security solution. However, cryptography has its costs. It takes considerably more computing power, and this is multiplied by several factors in the case of a database. The biggest difference between the various methods of database encryption is granularity.[8–10] Some methods encrypt a tuple, some a relation, and some the whole database.

Cryptography greatly affects database performance because each time a query is run, a large amount of data must be decrypted. Running queries is the database's primary purpose, thus decryption operations quickly become excessive. Although much research has focused on techniques to run queries on an encrypted database,[1,6–10] we have yet to achieve solid and acceptable efficiency and performance. This is true for relational DaaS databases, which are admittedly the extreme case of the database encryption requirements.

Early approaches used extensions to the query language that simply applied encryption before writing a value and running a decryption function before reading a value.[8] These types of extensions still exist in many commercial databases today, along with more advanced counterparts.[7,9] Even these counterparts are based on an absolute trust of the DBMS. In a cloud environment, this trust is absent. Some proposed alternative solutions include binning techniques, privacy homomorphism encryption,[5] early stopping comparisons, and separate encrypted indexes.[10] However, each solution has its own compromises and downsides: some involve the security of the data against certain attacks and others involve the operations available to the customer. For example, homomorphic encryption is effective for summations on encrypted values but prohibitively slow for other operations, such as joins.

Despite its severe performance limitations,[3,9] few alternatives exist to using encryption to protect data in a database. Divyakant Agrawal and his colleagues developed one possible method.[3] The idea began as a way to ensure privacy by splitting data between multiple hosts that can't communicate with each other (see Figure 1). Only the owner, who can access the hosts, can collect and combine the separate datasets to recreate the original. Agrawal and his colleagues extended this idea to apply to outsourced databases. They proposed that the bits that make up one piece of data be divided mathematically so it's impossible to infer the original data from either piece. This method is extremely fast compared to encryption but requires at least two separate—but homogenous—service providers. Unfortunately, it doesn't provide a clear way to protect the metadata (such as field names), although further extension of the technique could possibly address this.

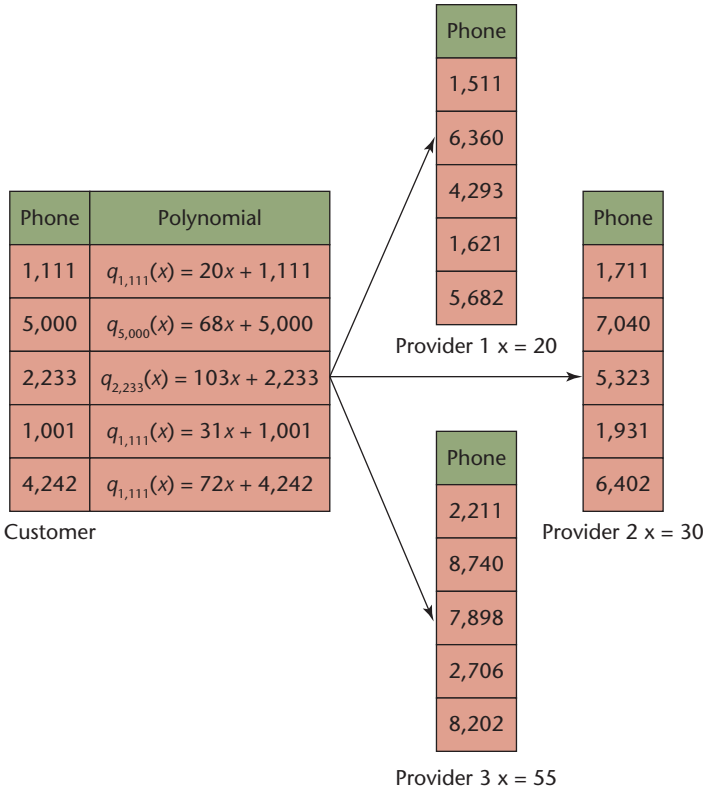

Figure 1. Dividing data.[3] The phone attribute data gets distributed among three service providers using secret sharing. The polynomials are generated using a different secret for each service provider.

Key Management

Naturally, if encryption is necessary to store data in the cloud, then the encryption keys can't be stored there. The customer must manage and control any key management system for any cryptographic method. For simple encryption schemas, this might not be an issue, but almost any real database requires a more complex system. Oddly enough, this might take form as a small database. Similarly, for tokenization schemas, this means maintaining a secure local database, which again, defeats the purpose of moving the original database to the cloud. Some schemas require a secure connection directly between the key management system and the cloud database. Clearly, this type of situation requires special arrangements with the service provider. Recent research uses two-level encryption that successfully allows the key management system to be stored in the cloud.[6] Although this is efficient, it hasn't yet been applied specifically to database encryption.

Authenticity (Integrity and Completeness)

In common database operations, users with limited rights might want to access a subset of data, and in the context of DaaS, they might also want to verify that the delivered results are valid and complete—that is, not poisoned, altered, or missing anything. A common approach to such a problem is to use digital signatures. The problem with digital signatures is that users typically don't have access to the data superset, so they can't verify any subset of data even if they're provided with the superset's digital signature. Too many possible subsets exist to maintain a digital signature for each; the same problem arises when the signatures are applied at the finest granularity level. In recent years, researchers have tried to find a solution to this problem.[5]

The primary proposal is to provide customers with the superset's signature and some metadata along with the query results. This metadata, called verification objects, lets customers fill in the blanks of the data to which they don't have access, and thus still validate the signature. There are two primary variations of this idea. One is based on Merkle trees—a concept created to sign multiple messages using binary hash trees—and the other is based on signature aggregation.

Data completeness is often left to the realm of authenticity and integrity, but in the case of DaaS, this topic needs special attention owing to the inherent lack of trust. Indeed, the methods for showing completeness are basically the same as those used for authenticity. They also use Merkle trees or aggregate signatures but include boundary records to aid in verification.

Other Issues

An unmentioned yet understood amount of security exists when a business hosts its own IT infrastructure. It's in the machines themselves, the servers used for computation. They're usually built by the employees or bought commercially and physically secured. Most important, the business generally knows and restricts what's running on its machines. This provides some assurance of certain things we take for granted until our data or application is out in the cloud.

Typically, in the cloud, likely even for DaaS, the business's data and applications are stored or running inside a virtual machine. This means that the virtual machine is probably running on a server with other virtual machines, some of which could be malicious. Although attacks against, with, and between virtual machines aren't known to exist in the wild, researchers have shown they're possible.[12]

Assuming that such attacks don't exist and that an application isn't running in a virtual machine, the hardware then becomes an issue. Modern servers are typically multiprocessor and probably multicore. Today's research shows it's possible for information to flow between cores on modern processors, meaning that a secure application running on one core might unknowingly pass information to a process on another core.

It gets worse when it comes to memory. Multicore processors often have complex caches, which makes the multicore problem worse. This creates a big issue when decrypting in the cloud. If a value is decrypted in the cloud, even just for a moment for comparison, it exists unencrypted in the memory of some machine. Because of the nature of the cloud, we not only don't know where this machine is, we also don't know what else is running on the machine. It's possible that another malicious cloud user is monitoring the machine's memory and reads our data or that the provider itself is monitoring the contents of the memory.

Because the server isn't under the customer's control and it's likely that the customer doesn't know anything about the server, it's obvious to question it. It's natural to question what malicious code is running on the server, whether the service provider is even aware, and who is running the code. However, the likelihood of these hardware attacks is very small. With advances in virtual malware, this might change in the future, but for now it remains a minor risk. In addition, these issues generally concern high-assurance systems, which would face many larger threats from a cloud computing environment, such as typical Internet attacks.

Each cloud service has its issues, and the majority of them are security related. Although cloud computing must overcome many obstacles before its widespread adoption, it can work now, in the right conditions. If businesses are careful about the questions they ask, demand a decent SLA, and understand the compromises, there's no reason they can't begin enjoying cloud services today. As the DaaS example shows, some issues and performance concerns exist, but the vast majority of technologies can be explored. For those more difficult technologies, such as DaaS, some database encryption and querying breakthroughs might be necessary before they're really viable. Even so, we probably won't have long to wait.

The rest boils down to trust. The service provider creates trust by creating a decent SLA with the services and processes available to back up its assurances. Customers measure their trust by evaluating risk. Only by understanding the risks they're taking can they understand the level of trust they can expect to have in any service provider. When customers have the right level of expectations and the insecurities are deemed manageable, cloud computing will finally gain ground and take hold as a usable technology. □

References

1. R. Sion, "Query Execution Assurance for Outsourced Databases," *Proc. 31st Int'l Conf. Very Large Data Bases* (VLDB 05), VLDB Endowment, 2005, pp. 601–612.
2. W. Lehner and K.-U. Sattler, "Database as a Service (DBaaS)," *IEEE 26th Int'l Conf. Data Engineering* (ICDE 10), IEEE CS Press, 2010, pp. 1216–1217.
3. D. Agrawal et al., "Database Management as a Service: Challenges and Opportunities," *IEEE 25th Int'l Conf. Data Engineering* (ICDE 09), IEEE CS Press, 2009, pp. 1709–1716.
4. M. Armbrust et al., "A View of Cloud Computing," *Comm. ACM*, vol. 53, no. 4, 2010, pp. 50–58.
5. E. Ferrari, "Database as a Service: Challenges and Solutions for Privacy and Security," *IEEE Asia-Pacific Services Computing Conf.* (APSCC 09), IEEE CS Press, 2009, pp. 46–51.
6. S. de Capitani di Vimercati et al., "Encryption Policies for Regulating Access to Outsourced Data," *ACM Trans. Database Systems*, vol. 35, no. 2, article 12, 2010.
7. H. Hacigumus et al., "Executing SQL over Encrypted Data in the Database-Service-Provider Model," *Proc. ACM SIGMOD Int'l Conf. Management of Data* (SIGMOD 02), ACM Press, 2002, pp. 216–227.
8. H. Hacigumus, B. Iyer, and S. Mehrotra, "Providing Database as a Service," *Proc. 18th Int'l Conf. Data Engineering* (ICDE 02), IEEE CS Press, 2002, pp. 29–38.
9. J. Hu and A. Klein, "A Benchmark of Transparent Data Encryption for Migration of Web Applications in the Cloud," *Proc. 8th IEEE Int'l Conf. Dependable, Autonomic, and Secure Computing* (DASC 09), IEEE CS Press, 2009, pp. 735–740.
10. E. Shmueli et al., "Database Encryption: An Overview of Contemporary Challenges and Design Considerations," *SIGMOD Record*, vol. 38, no. 3, 2010, pp. 29–34.
11. F. Chang et al., "Bigtable: A Distributed Storage System for Structured Data," *ACM Trans. Computing Systems*, vol. 26, no. 2, article 4, 2008.
12. R. Chow et al., "Controlling Data in the Cloud: Outsourcing Computation without Outsourcing Control," *Proc. ACM Workshop Cloud Computing Security* (CCSW 09), ACM Press, 2009, pp. 85–90.
13. L.J. Sotto, B.C. Treacy, and M.L. McLellan, "Privacy and Data Security Risks in Cloud Computing," *15 Electronic Commerce & Law Report*, BNA, 3 Feb. 2010, pp. 186–188.
14. M. Jensen et al., "On Technical Security Issues in Cloud Computing," *IEEE Int'l Conf. Cloud Computing*, IEEE CS Press, 2009, pp. 109–116.
15. B.R. Kandukuri, R. Paturi V, and A. Rakshit, "Cloud Security Issues," *IEEE Int'l Conf. Services Computing* (SCC 09), IEEE CS Press, 2009, pp. 517–520.

Joel Weis *is a graduate student at the University of Idaho and recipient of the Federal Cyber Service Scholarship for Service. His research interests include cybersecurity, information assurance, cloud technologies, databases, and embedded systems. Weis has a BS in computer science from the University of Hawaii at Hilo. Contact him at Weis.Joel@gmail.com.*

Jim Alves-Foss *is director of the University of Idaho's Center for Secure and Dependable Systems and a professor of computer science. His research interests include design and analysis of secure distributed systems, with a focus on formal methods and software engineering. Alves-Foss has a PhD in computer science from the University of California, Davis. Contact him at jimaf@uidaho.edu.*

cn *Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.*

Call for Papers

Lost Treasures of Computer Security & Privacy

for *IEEE Security & Privacy*'s November/December 2012 issue

NEW DEADLINE: final submissions due to ScholarOne by 1 March 2012

Please email the guest editors (sp6-2012@computer.org) a brief description of the article you plan to submit by 1 February 2012.

Since the release of the "Anderson Report" (Computer Security Technology Planning Study, October 1972), researchers have published tens of thousands of computer security papers. With the plethora of research needed to advance computer security, we must learn from the past to avoid wasted effort. Unfortunately, key security insights often remain hidden among rambling technical reports or obscured by government policies and regulations that have fallen from favor. This special issue addresses key lessons from the past 50 years—not merely to recapitulate them, but to learn from them.

View the full call for papers and submission link at www.computer.org/security/cfp.