UNIVERSITY OF CAMBRIDGE

Security and privacy at Cambridge (1/8)

The University of Cambridge is home to some of the world's leading computer security researchers, with a long history of key contributions to the field. Cambridge's early interests in security include the identification of large primes (Wheeler 1949), one-way password encryption (Needham 1962), the capability system security model (Wilkes, Needham, Walker 1970–1977), the Needham-Schroeder Protocol (1978), and the Burrows-Abadi-Needham logic (1989). We continue to make core contributions in the field of computer security – cryptographic protocol design, CPU and operating system security, anonymity research, malware analysis – but also foundational cross-disciplinary work in security economics, cybercrime measurement, security psychology, human factors, and domestic and international policy.

Security and privacy research spans many groups at Cambridge, exploring issues ranging from CPU security to cybercrime:

- Computer Laboratory - Security Research Group
- Computer Laboratory - Network and Operating System Group
- Computer Laboratory - OPERA Group (distributed systems)
- Computer Laboratory - Computer Architecture Group
- Computer Laboratory - Programming, Logic, and Semantics Group
- Computer Laboratory - Digital Technologies Group (DTG)
- Centre for Science and Policy (CSaP)

Full-time academic staff with security or privacy research focuses are Ross Anderson, Jean Bacon, Alastair Beresford, Jon Crowcroft, John Daugman, Steven Hand, Markus Kuhn, Simon Moore, Larry Paulson, and Frank Stajano. Post-doctoral researchers include Jonathan Anderson, David Chisnall, Richard Clayton, Khilan Gudka, Steven Murdoch, Michael Roe, Sergei Skorobogatov, and Robert Watson.

Computer Laboratory computer security group

The security research group consists of three full-time faculty members, six full-time post-doctoral researchers, twelve PhD students, and several master students working on security-focussed dissertations. The group maintains dedicated research facilities including a Tamper Lab for reverse engineering hardware, side-channel and fault injection attacks, and analysing electromagnetic emanations from computing devices. The security group works closely with other groups, bringing a security perspective to their projects. The group has a web site and blog, which carry our recent research, musings on security, and job/studentship ads:

http://www.lightbluetouchpaper.org/
http://www.cl.cam.ac.uk/research/security/

Active security and privacy research areas

- Anonymous communication
- API and protocol security
- Application compartmentalisation techniques
- Authentication and biometric identification systems
- Banking and payment system security
- Capability systems
- Compromising emanations
- Cryptology
- Digital forensics
- Distributed system and cloud computing security
- Economics of information security and cybercrime
- Formal methods
- Hardware security
- Location and positioning systems
- Malware analysis
- Medical information security
- Mobile and embedded system security
- Operating system security
- Passwords
- Privacy and freedom issues
- Programming language security
- SCADA and the security of industrial control systems
- Security and human behaviour
- Security protocols
- Social networking and privacy
- Steganography
- Tampering with tamper-resistant devices
- Temporal security properties

Security and privacy teaching

The Computer Laboratory's undergraduate and masters programmes provide in-depth teaching of computer security foundations and current research; operating systems, networking and programming languages courses also necessarily consider security. Outside of the classroom, students have access to practical competitive exercises to break protection and anonymity systems in order to concretely understand software and protocol security. Several undergraduates and masters students write security-related dissertations or essays each year. Masters and PhD students frequently publish security-related research.

Undergraduate degree in Computer Science

- *Part I Security* is an introductory course providing every student with the basics in security. Material includes the cryptography, protocols, programming language, application, and operating system security.
- *Part II Security* is an advanced course teaching about security policy, security usability, security economics, security protocols, cryptography, hardware security, privacy, anonymity and concurrency vulnerabilities.

MPhil in Advanced Computer Science (ACS)

- *Principles and Foundations of Security* is a research readings course reviewing key historic security texts and themes in security research, including tensions between attack and defense, human factors, security economics, reasoning protocols, programming language security, and the evolution of access control and protection models.
- *Current Application Research in Security* is a research readings course exploring contemporary research areas, including supply-chain trojans in hardware, malware analysis, secure processor design, and banking system security.

PhD in Computer Science

The PhD in Computer Science mentors students in award-winning research and methodology. Recent security-related PhDs include:

- *Privacy engineering in social networks*
- *Guessing human-chosen secrets*
- *Robust security for the electricity network*
- *Complex network analysis for secure and robust communications*
- *Verification of security protocols based on multicase communication*
- *Distributed virtual environment scalability and security*
- *New approaches to operating system security extensibility*
- *Active electromagnetic attacks on secure hardware*

UNIVERSITY OF CAMBRIDGE

Security economics, policy, and psychology (2/8)

Security economics

Ross Anderson, Joseph Bonneau, Mike Bond, Richard Clayton, Steven Murdoch

An early observation that complex systems fail because the incentives are wrong led to Cambridge founding the field of security economics, which has 100+ researchers worldwide. This new, and highly successful, approach takes the view that it is usually more relevant to look at the economics of a system than at the computer science when seeking to understand whether or not it will be secure.

Two major reports have been written for ENISA, on applying security economics to network security (2008) and on the resilience of the Internet (2011) where we explain how network operators negotiate peering and transit, what goes wrong, how they deal with failures and where the incentives for resilience are inadequate.

Research into cybercrime, particularly “phishing”, has used econometric approaches (analysis of website lifetimes for example) to explain the variations in the effectiveness of countermeasures to criminal activity. Along similar lines, a joint project with the National Physical Laboratory (NPL), partly funded by the EPSRC, is developing robust measurements of Internet security mechanisms.

Other research in this field has looked at website data collection policies, password strength, public policy on malware removal, app security and the security economics of electricity ‘smart meters’.

Measuring cybercrime

Richard Clayton, Ross Anderson

We first measured “phishing” in 2007, showing that the newly introduced ‘fast flux’ techniques were allowing the criminals to keep their fake banking websites running for longer. The industry told us that the average lifetimes we observed were higher than they expected – we realised that we knew about sites that the banks did not and, not surprisingly, websites they hadn't been told about didn't get taken down. Our data about phishing websites was better than the banks because competing take-down companies would share information with us, but not with each other – we showed for the first time the true cost of this data hoarding.

We've continued to work on phishing, but have also been publishing measurement work on spam fighting, and recently on High Yield Investment Programmes – Ponzi schemes that pay outrageously high returns to investors from the money coming in from newer victims.

In 2012 we led a team of international experts to create the first systematic study of the costs of cybercrime. We found that the direct costs of traditional offences such as tax and welfare fraud (now mainly done “online”) cost the typical citizen in the low hundreds of pounds a year; transitional frauds (reinvented for cyberspace) cost a few pounds; while the new cyberspace-only crimes cost in the tens of pence. However, the indirect costs and defence costs are much higher for transitional and new crimes. Our figures suggest that we should spend less in anticipation of cybercrime (on antivirus, firewalls, etc.) and more in response – on the prosaic business of hunting down cybercriminals and throwing them in jail.

Understanding the psychology of scams

Frank Stajano (joint work with Paul Wilson of BBC3's The Real Hustle)

Security engineers build system defences, but real users don't follow engineer logic. Result: systems are vulnerable to attack. How can we understand what makes users vulnerable?

Our approach: learn from the fraudsters! They understand the victims’ psychology better than most security engineers.

We distilled seven general principles by documenting and analysing hundreds of observed scams. Knowledge of these principles can be used to strengthen system security.

The viewpoint that “it's the fault of those gullible users” is arrogant and idiotic. Some behavioural patterns that make us vulnerable to fraud are just human nature. It is up to smart system designers to prevent their exploitation.

Frank Stajano and Paul Wilson, “Understanding scam victims: seven principles for systems security”, Communications of the ACM, 54(3):70-75, March 2011.

UNIVERSITY OF CAMBRIDGE

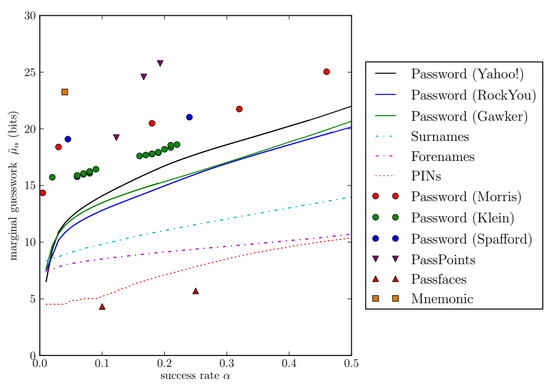
Authentication and identity (3/8)

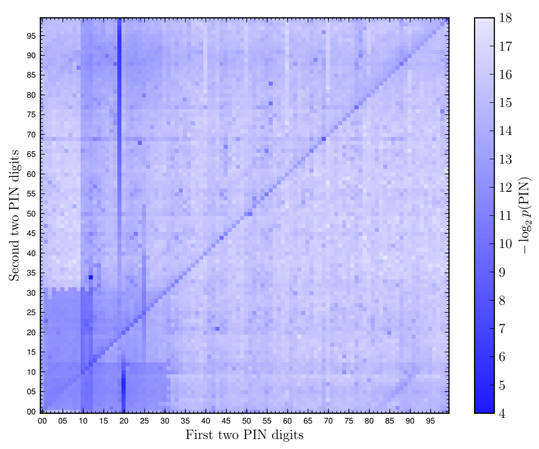
Human authentication in practice

Joseph Bonneau, Ross Anderson, Sören Preibusch

Secure and usable authentication remains elusive: decades of research have failed to move the world away from basic textual passwords. Yet, surprisingly little research attempted to answer basic questions such as: are human-chosen PINs or passwords easier to guess? What about personal knowledge questions such as “what is your mother's maiden name?” Do some users tend to pick better passwords than other users?

Recent research at Cambridge has begun to provide both sound methodology to answer these questions and reliable estimates from a collection of massive data sets. This effort has required collecting the largest-ever corpus (70 M) of human-chosen passwords in a privacy-preserving manner, conducting the first large-scale surveys of human PIN choice, and collecting the world's largest corpora of human names (over 250 M).



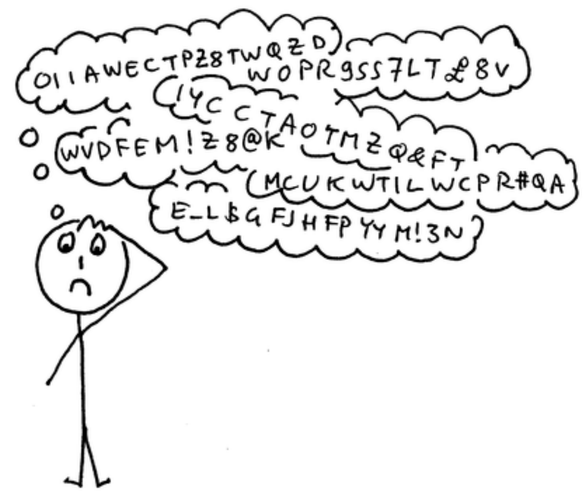


This project has provided some surprises, like older users picking stronger passwords than younger users or password strength policies making little impact on guessing difficulty. It has also provided some compelling numbers for security engineers to remember. Passwords are about equal to 10-bit random strings, or 3-digit decimal numbers, against a rate-limited attacker. If a thief steals a wallet with an ATM card and an ID listing date of birth, she has about a 10% chance of guessing the PIN.

Beyond passwords

Frank Stajano

Users are told by security people that their passwords must be unguessable, must contain mixed-case letters, numbers and symbols, must not be written down, must all be different and must also be changed every couple of months. The intersection of all these constraints is the empty set. It's objectively impossible to follow all these directives at once.



As the number of online accounts per person keeps growing, passwords are not sustainable as a user authentication scheme. What shall we do long term? We are researching a future alternative called Pico, based on hardware tokens: no secrets to memorize, scales to thousands of accounts. Besides usability, it also solves many security problems: resisting brute-forcing, eavesdropping, phishing, keylogging etc. Pico unlocks in the presence of its owner by recognizing miniature gadgets embedded in wearable items such as clothes and jewellery (Picosiblings). The user never has to type a PIN to unlock the Pico.

Pico requires major changes to infrastructure and does not aim to replace passwords overnight. It looks at where we might wish to be in a decade or two. We know we'll have to improve on passwords eventually.

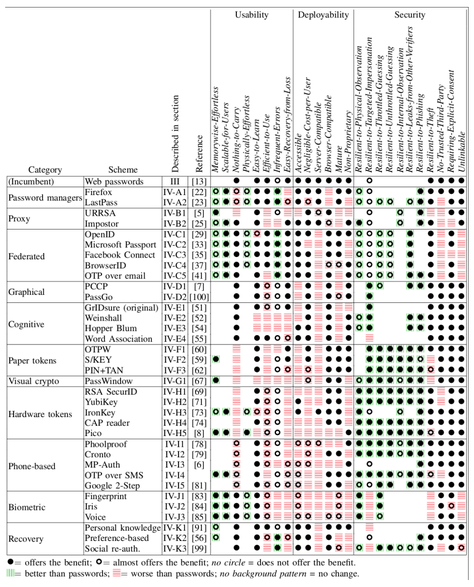
The Pico design was peer-reviewed, published, presented at conferences in three continents and eventually awarded a competitive European Research Council grant worth over £1M to produce a reference implementation.

Frank Stajano, “Pico: no more passwords!" In Proc. Security Protocols Workshop 2011, LNCS 7114, Springer 2011.

A framework for comparative evaluation of password replacement schemes

Frank Stajano, Joseph Bonneau, Cormac Herley (Microsoft), Paul van Oorschot (Carleton)

Passwords have well-known security and usability problems. Over the past couple of decades, dozens of alternative schemes were proposed. Why, then, do we still use passwords so extensively? Don’t the suggested replacements offer any improvements? This project offered a structured and well-researched answer.



Category	Scheme	Described in section	Reference	Usability: Memorywise-Effortless	Usability: Scalable-for-Users	Usability: Nothing-to-Carry	Usability: Physically-Effortless	Usability: Easy-to-Learn	Usability: Efficient-to-Use	Usability: Infrequent-Errors	Usability: Easy-Recovery-from-Loss	Deployability: Accessible	Deployability: Negligible-Cost-per-User	Deployability: Server-Compatible	Deployability: Browser-Compatible	Deployability: Mature	Deployability: Non-Proprietary	Security: Resilient-to-Physical-Observation	Security: Resilient-to-Targeted-Impersonation	Security: Resilient-to-Throttled-Guessing	Security: Resilient-to-Unthrottled-Guessing	Security: Resilient-to-Internal-Observation	Security: Resilient-to-Leaks-from-Other-Verifiers	Security: Resilient-to-Phishing	Security: Resilient-to-Theft	Security: No-Trusted-Third-Party	Security: Requiring-Explicit-Consent	Security: Unlinkable
(Incumbent)	Web passwords	III	[13]	•	•	•	•	•	•	•	•	•	•	•	•	•	○											
Password managers	Firefox	IV-A1	[22]	•	•	○	•	•	•	•	•	•	•	•	•	○	○									●	●	
	LastPass	IV-A2	[23]	•	•	○	•	•	•	•	•	•	•	•	•	○	○									●	●	
Proxy	URRSA	IV-B1	[5]	•			•		○			•	○	○				○										
	Impostor	IV-B2	[25]	•			•		○			•	○	○				○										
Federated	OpenID	IV-C1	[29]	•	•	○	•	•	○	•	•	•	○	•	•			○	○							●	●	
	Microsoft Passport	IV-C2	[33]	•	•	○	•	•	○	•	•	•	○	•	•			○	○							●	●	
	Facebook Connect	IV-C3	[35]	•	•	○	•	•	○	•	•	•	○	•	•			○	○							●	●	
	BrowserID	IV-C4	[37]	•	•	○	•	•	○	•	•	•	○	•	•			○	○							●	●	
	OTP over email	IV-C5	[41]	•	•	○	•	•	○	•	•	•	○	•	•			○	○							●	●	
Graphical	PCCP	IV-D1	[7]				•		○					•				○										
	PassGo	IV-D2	[100]				•		○					•				○										
Cognitive	GrIDsure (original)	IV-E1	[51]				•		○					•														
	Weinshall	IV-E2	[52]				•		○					•				○										
	Hopper Blum	IV-E3	[54]				•		○					•				○										
	Word Association	IV-E4	[55]				•		○					•														
Paper tokens	OTPW	IV-F1	[60]				•		○					•														
	S/KEY	IV-F2	[59]	•			•		○					•														
	PIN+TAN	IV-F3	[62]				•		○					•														
Visual crypto	PassWindow	IV-G1	[67]	•			•		○					•														
Hardware tokens	RSA SecurID	IV-H1	[69]				•		○					•														
	YubiKey	IV-H2	[71]				•		○					•														
	IronKey	IV-H3	[73]	•			•		○					•														
	CAP reader	IV-H4	[74]				•		○					•														
	Pico	IV-H5	[8]	•			•		○					•														
Phone-based	Phoolproof	IV-I1	[78]				•		○					•														
	Cronto	IV-I2	[79]				•		○					•														
	MP-Auth	IV-I3	[6]				•		○					•														
	OTP over SMS	IV-I4		•			•		○					•														
	Google 2-Step	IV-I5	[81]				•		○					•														
Biometric	Fingerprint	IV-J1	[83]	•			•		○					•														
	Iris	IV-J2	[84]	•			•		○					•														
	Voice	IV-J3	[85]	•			•		○					•														
Recovery	Personal knowledge	IV-K1	[91]				•		○					•														
	Preference-based	IV-K2	[56]				•		○					•				○										
	Social re-auth.	IV-K3	[99]				•		○					•														

●= offers the benefit; ○= almost offers the benefit; *no circle* = does not offer the benefit.
▒ better than passwords; ▒ worse than passwords; *no background pattern* = no change.

We build a large 2D matrix. Across the columns we define a broad spectrum of 25 potential benefits in the areas of usability, security and deployability. Next, in the rows, we identify 35 representative schemes covering 11 broad categories. We then rate each scheme individually on whether it offers each benefit. The resulting matrix allows readers to compare features at a glance and to recognize general patterns.

Contrary to the optimistic claims of scheme authors, who often completely ignore some evaluation criteria, none of the examined schemes does better than passwords when rated on all 25 benefits of this objective benchmark.

Many people repeat the mistakes of history because they didn’t understand the history book. Here, we had to write one first! As pointed out during peer review, this work is a foundational starting point for further research in the area and a useful sanity check for future password replacement proposals.

J. Bonneau, C. Herley, P.C. van Oorschot, F. Stajano (lead author). “The quest to replace passwords: a framework for comparative evaluation of password replacement schemes.” In Proc. IEEE Security & Privacy 2012.

The management of identity

David Evans (Derby), Jean Bacon, Alastair Beresford

Future applications will need to identify places and objects, as well as people. We are interested in extending the traditional model of identity – one or more for each individual – to these scenarios. Here we bind context to person or a group, leading to a pseudonym that is linkable only by those knowing the context. This context might be an attribute of the physical location (such as bus number plates, pub names, or street names) or knowledge that is dependent on being in a particular location (e.g., the number of people standing on the pavement outside John Lewis).

A pseudonym for person *i* linkable by person *j* in context *n* is

$$P_{ij}^{(n)} = H_{K_{ij}}(\mathrm{ID}_i, C_n)$$

where K_{ij} is a key known only to users *i* and *j* and $H_K\langle\cdot\rangle$ is a message authentication code with key *K*. For a set of pseudonyms $G = \{P_1, P_2, \ldots\}$, $P_G = \langle P_1, P_2, \ldots\rangle_K$ where $\langle\cdot\rangle_K$ is a privacy-preserving set representation such as that by Hohenberger and Weis having key *K*.

Pseudonym Properties

- Given access to *P*, an attacker without the correct key cannot infer the ID or the context of the corresponding person.
- Given $P_{ij}^{(n)}$ and $P_{ij}^{(m)}$, user *j* can link the movement of user *i* from context C_n to C_m.
- Given access to $P_{ij}^{(n)}$, $P' = P_{ij}^{(m)}$, but not K_{ij} an attacker cannot determine that *P* and *P′* represent the same person.
- Given access to P_G and *K*, a user can check whether specific pseudonyms are in *G*.
- Given P_G and K, a user knowing the keys to some of the pseudonyms in G learns nothing about the status of pseudonyms whose keys are unknown.
- Unrestricted distribution of individual or group pseudonyms does not compromise real-world identity.

UNIVERSITY OF CAMBRIDGE

Hardware security and signal analysis (4/8)

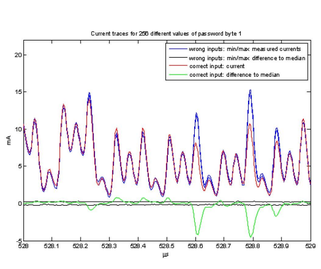
Hardware tamper resistance

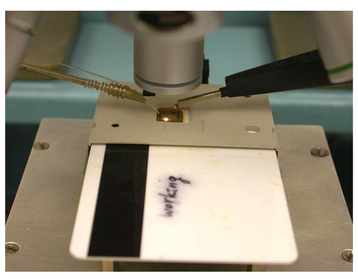
Markus G. Kuhn, Sergei Skorobogatov

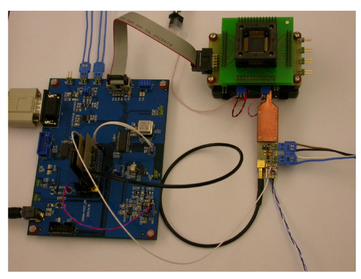
Protection against physical attacks has become an essential part of many modern system designs. These days we have a continuous battle between manufacturers who invent new security solutions learning from previous mistakes, and a hacker community that is constantly trying to break protection in various devices. The importance of security is dictated by the amount of valuable and sensitive information stored on the chip. This could be cryptographic keys, secret data, company secrets, intellectual property, electronic money or banking smartcards.

Our group have invented semi-invasive attacks (optical fault injection) which have forced the industry to rethink protection mechanisms in smartcards and amend Common Criteria requirements. As with invasive attacks (microprobing, chip modification), they require opening the chip in order to get access to its surface without destroying it or creating contacts to internal wires. Those attacks are as easy to implement as inexpensive non-invasive attacks (power analysis, glitching).

In collaboration with Quo Vadis Labs we developed a new side-channel analysis technique. This breakthrough approach means it is now possible to extract encryption keys from devices and systems up to a million times faster than state-of-the-art power analysis techniques, e.g. DPA. Our recent research is focused on Hardware Assurance – testing of silicon chips for backdoors and trojans. Because of the speed at which analysis can be performed, it becomes possible to identify hidden backdoors and trojans in silicon chips – a task that is not feasible with current DPA methods.





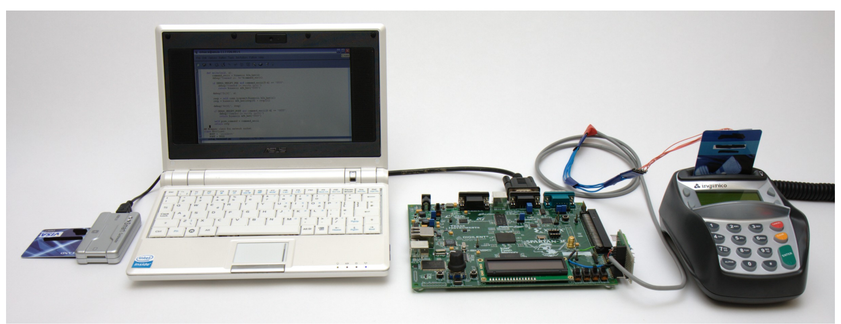


Banking and payment system security

Ross Anderson, Mike Bond, Omar Chourary, Steven Murdoch, Laurent Simon

Known in the UK as “Chip and PIN”, EMV (Europay, MasterCard, Visa) is the dominant standard for smart-card payments worldwide. Introduced to reduce card fraud, EMV is used throughout Europe; it is being introduced in the US, Canada and South America. EMVCo estimates that over a billion EMV payment cards are in circulation. EMV makes card transactions more secure by adding a chip to cards to make them harder to counterfeit and requiring customers to enter a PIN to authorize payment. While initially reducing fraud, criminals adapted to the change, resulting in increased losses.

Research at Cambridge has discovered numerous vulnerabilities in the deployed Chip and PIN system, including the ability for criminals to trick terminals into accepting an incorrect PIN for a stolen card, failures in the tamper resistance measures present in widely deployed Chip and PIN terminals, and ways for corrupt bank employees circumvent protections against insider attacks to discover customer PINs. We have developed methods to resolve these vulnerabilities and work with industry to have these improvements deployed. Smart Architects Ltd sell auditing equipment, developed at Cambridge, to detect these vulnerabilities. Methods for securing online banking against attack by man-in-the-middle malware, developed at Cambridge, are being commercialised by a spin-out company (Cronto Ltd) and are in use at several banks.

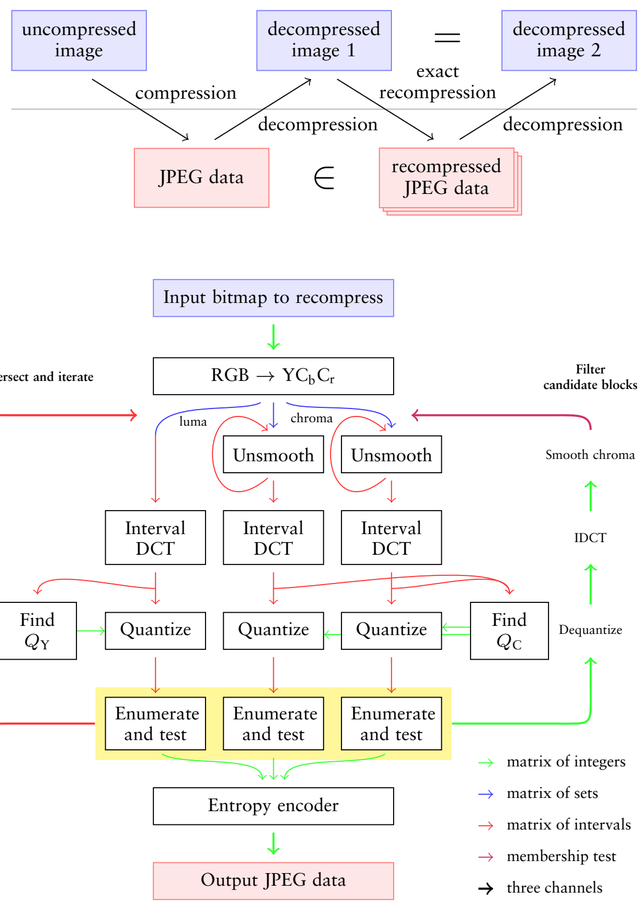


Forensic signal analysis

Markus G. Kuhn, Andrew Lewis

Widespread use of digital cameras, compression, high-capacity storage, and advanced tools (video editors, SDR, etc.) pose new challenges to forensic investigators, who need techniques to confirm claimed origins and processing histories of signal evidence, and recover data without the cooperation of the hardware owner or designer.

In collaboration with the Metropolitan Police, we designed a sophisticated tool for recovering compressed video data from fragmented storage where file-system metadata is unavailable. Manufacturers are unwilling to release documentation for proprietary file-systems, block allocation, and Flash wear-leveling algorithms, leaving investigators with a random puzzle of 4KB memory blocks to be assembled back into a video stream (H.264, etc.), a task that our new high-performance parser can handle for gigabytes of data.

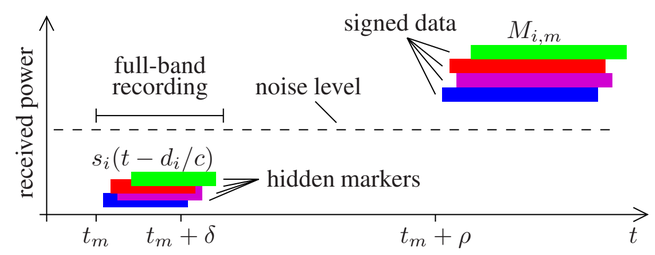


Our *exact JPEG recompressor* uses iterative interval arithmetic to invert a JPEG decompressor, recreating the original compressed bitstream. It can help to identify/exclude particular decoders as a source, but also has applications in evaluating copy-protection schemes.

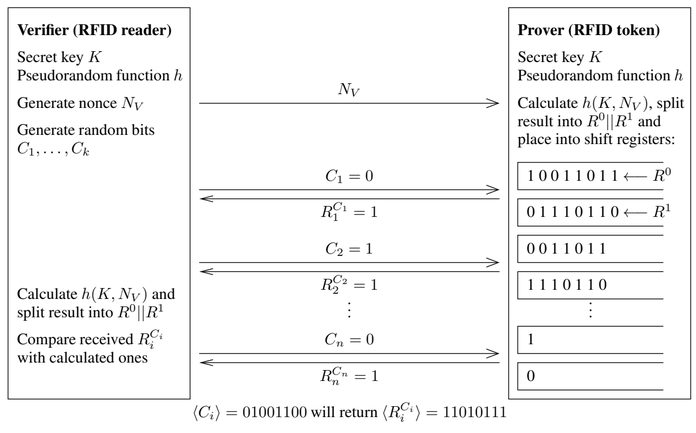
Location security: cryptography at the speed of light

Markus G. Kuhn

Global navigation satellite systems still lack signal-integrity protection for mass-market applications. A particular challenge is the need to protect not only the integrity of broadcast data, but also the nanosecond-accurate relative arrival times of signals from different satellites, from which receivers determine their position. We have proposed several new types of signal-authentication and spoofing detection techniques. A steganographic transmission scheme with delayed release of spreading keys provides the same asymmetric security that made digital signatures so useful: the ability to verify a signal does not lead to the ability to spoof it. This is of particular importance in civilian mass-market applications where the holder of the receiver may want to manipulate its reading: road toll, pay-as-you-drive car insurance, or offender-tagging. In addition, we have also proposed heuristics that help GPS receivers detected spoofed signals.



Distance-bounding protocols ascertain both the identity of a communications partner and also their location. They securely answer questions such as “is this smartcard really within 1 metre of the reader?”, by incorporating a rapid exchange of single challenge-response bits, where the speed of light is the main part of the round-trip time. We proposed the first distance-bounding protocol optimized for RFID applications with noisy wide-band channels.



UNIVERSITY OF CAMBRIDGE

Systems software security (5/8)

Systems security research

Robert N. M. Watson, Jonathan Anderson, Ross Anderson, Jean Bacon, David Chisnall, Jon Crowcroft, Khilan Gudka, Steven Hand, Wei Ming Khoo, Hyoungshick Kim, Ben Laurie (Google), Pietro Liò, Anil Madhavapeddy, Steven Murdoch, Michael Roe, Rubin Xu

Operating systems (OSs) and programming language runtimes are central elements of trusted computing bases (TCBs)—the minimal subset of a system that must be correct for it to be secure. Cambridge is exploring many techniques to improve OS security, including virtualisation, new security models, blending of formal methods with engineering, and clean-slate redesigns. Recent research includes:

- **Aurasium** is an Android application sandboxing system that imposes security and privacy policies through application transformation.
- **Capsicum**, a Google-funded hybrid capability system, blends the historic capability security model with contemporary OS design. Capsicum supports application compartmentalisation, a key vulnerability mitigation technique, and appears in the FreeBSD OS.
- **CHERI**, a capability-extended RISC CPU supporting granular and highly scalable application compartmentalisation.
- **Mirage**, an RCUK-funded clean-slate OS, written in OCaml. Implementing the TCB in a type-safe, functional language offers significant security and reliability improvements through access to formal verification and immunity to many traditional vulnerabilities.
- **SOAAP**, a suite of analysis tools to assist in identifying and semi-automatically applying application compartmentalisation.
- **TESLA**, a framework for validating temporal security properties in software TCBs that cannot be checked using current “instantaneous” software assertion techniques.
- **Xen**, an EPSRC-funded hypervisor for full system virtualisation. Released as open source and now widely used for security and cloud computing systems such as Amazon EC2 and Rackspace.

Xen Capsicum CIEL

Phylogenetic-inspired techniques for malware analysis: attacks and defence

Wei Ming Khoo, Hyoungshick Kim, Pietro Liò

Today’s malware is written to be persistent. Financial incentives are the dominant motivation for writing and spreading malware, and making sure that the malware remains as long as possible on the victims’ machines. As a result of this, malware exist in families, often numbering in the thousands, in order to constantly evade antivirus products and operating systems defences. However, malware is seldom written from scratch. Because new malware variants are usually inspired by previous ones, at some level they show a convergence of functionality.

We developed a framework for abstracting, aligning and analysing malware execution traces and are exploring the use of state of the art phylogenetic methods, whose strengths lie in pattern recognition and visualisation, to derive the statistical relationships within contemporary malware families. Some of methods used include phylogenetic trees and networks, motifs, logos, composition biases, and tree topology comparison methods.

Sequence alignment algorithms have recently found a use in detecting code clones, software plagiarism, code theft, and detecting polymorphic malware. This approach involves extracting software birthmarks, in this case sequences, from programs and comparing them using sequence alignment, a procedure which has been intensively studied in the field of bioinformatics. While sequence alignment may cope well with accidental DNA and protein mutations, we have shown that it can be vulnerable to specific insertion and deletion schemes and to concurrent programming.

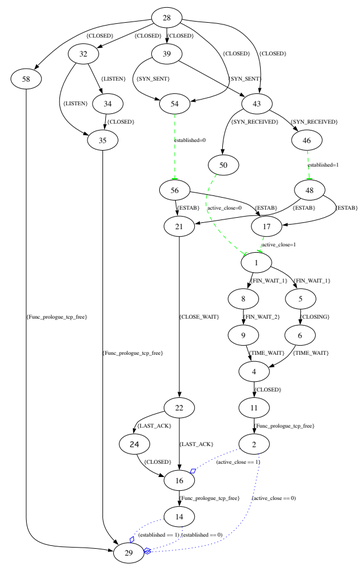
TESLA: temporal security assertion language

Robert N. M. Watson, Jonathan Anderson, Steven Hand, Anil Madhavapeddy, Ilias Marinos, Steven J. Murdoch, Michael Roe

Experienced programmers of complex software systems document and test invariants through extensive use of software assertions. Unfortunately, C language assertions are only able to test invariants that can be evaluated at the instant assert is invoked. Checking more complex temporal properties requires programmers to manually instrument code and data structures. This makes checking safety properties (e.g., correct memory allocation protocols, check-before-use, conformance to the TCP state machine, cryptographic protocol state machines, and wall clock time-liness goals) verbose, time-consuming, and error-prone. As most critical trusted computing bases (TCBs) are implemented in the C language, this presents a significant challenge in developing secure systems, whose security properties are frequently temporal properties untestable with C assertions.

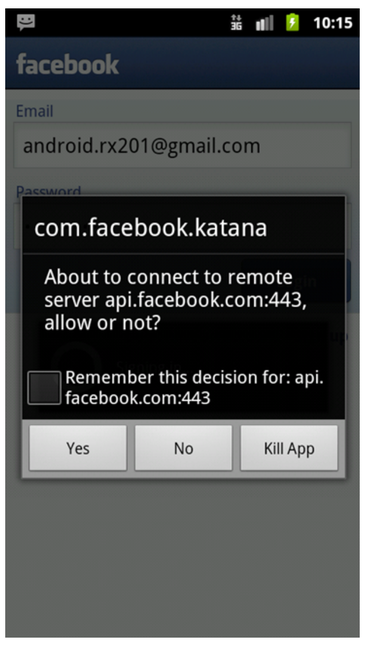
Temporally Enhanced Security Logic Assertions (TESLA) enhance the C language and runtime to support temporal assertions, which are able to reference past and future events. TESLA provides tools to help programmers better understand and enforce temporal security properties, in the short term for debugging and testing, but in the longer term, possibly also for live enforcement. It is particularly useful in the context of access control, where access control checks are an artefact of the underlying access control policy: TESLA can detect and report missing access control checks based on an abstract policy.

```
#include <netinet/tcp_fsm.h>

automaton tcp_connect() {

  void active_close(struct tcpcb *tp) {
    tp->t_state = TCPS_FIN_WAIT_1;
    either {
      tp->t_state = TCPS_CLOSING;
      tp->t_state = TCPS_TIME_WAIT;
    } or {
      tp->t_state = TCPS_FIN_WAIT_2;
      tp->t_state = TCPS_TIME_WAIT;
    }
    tp->t_state = TCPS_CLOSED;
    tcp_free(tp); exit;
  };

  void established(struct tcpcb *tp) {
    tp->t_state = TCPS_ESTABLISHED;
    either {
      active_close(tp->t_state);
    } or {
      tp->t_state = TCPS_CLOSE_WAIT;
      tp->t_state = TCPS_LAST_ACK;
      optional { tp->t_state = TCPS_CLOSED; }
    }
    tcp_free(tp); exit;
  };

  void main(struct tcpcb *tp) {
    tp->t_state = TCPS_CLOSED;
    either {
      tp->t_state = TCPS_LISTEN;
      optional { tp->t_state = TCPS_CLOSED; }
      tcp_free(tp);
    } or {
      optional { tp->t_state = TCPS_SYN_SENT; }
      either {
        tp->t_state = TCPS_SYN_RECEIVED;
        either {
          established (tp->t_state);
        } or {
          active_close(tp->t_state);
        }
      } or {
        established (tp->t_state);
      }
    } or {
      tcp_free(tp); exit;
    }
  }
}
```



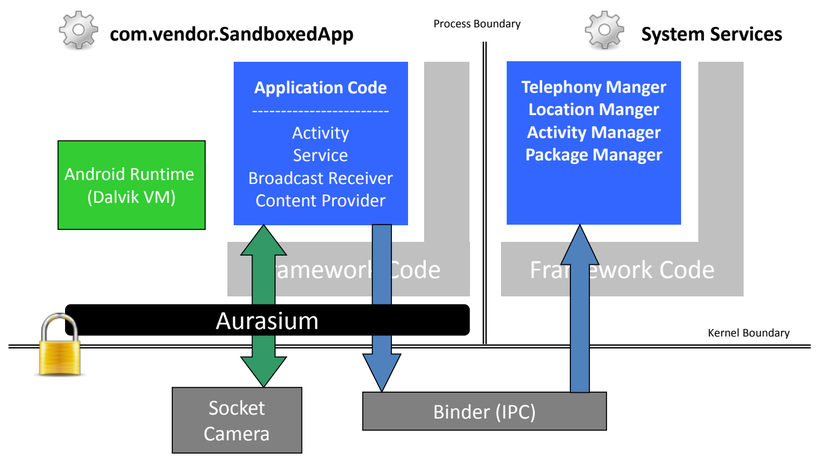
Aurasium: practical policy enforcement for Android applications

Rubin Xu, Hassen Saidi (SRI), Ross Anderson



The increasing popularity of Google’s mobile platform Android makes it the prime target of the latest surge in mobile malware. Most research on enhancing the platform's security and privacy controls requires extensive modification to the operating system, which has significant usability issues and hinders efforts for widespread adoption. We develop a novel solution called Aurasium that bypasses the need to modify the Android OS while providing much of the security and privacy that users desire. We automatically repackage arbitrary applications to attach user-level sandboxing and policy enforcement code, which closely watches the application’s behaviour for security and privacy violations such as attempts to retrieve a user's sensitive information, send SMS covertly to premium numbers, access malicious IP addresses and known privilege escalation attempts.

Experiments show that we can apply this solution to a large sample of benign and malicious applications from various sources with over 99% success rate, and without significant performance and storage overhead.



UNIVERSITY OF CAMBRIDGE

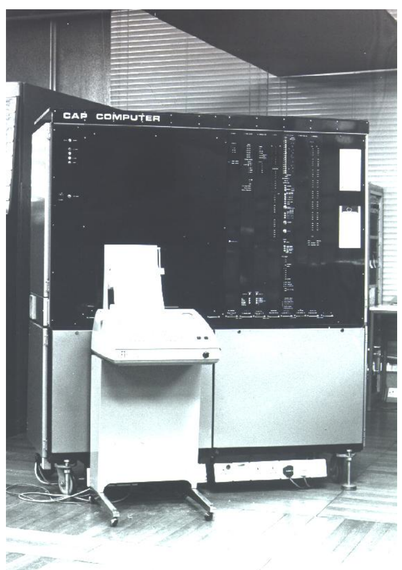
Hybrid capability systems (6/8)

Capability system research

Robert N. M. Watson, Jonathan Anderson, Ross Anderson, David Chisnall, Khilan Gudka, Steven Hand, Kris Kennaway (Google), Ben Laurie (Google) Simon W. Moore, Steven Murdoch, Robert Norton, Michael Roe, Jonathan Woodruff, Bjoern Zeeb

Programmers are increasingly turning to *compartmentalisation* to mitigate inevitable vulnerabilities: software is decomposed into many sandboxed components, each with only the rights it requires to operate. This approach exploits the *principle of least privilege*: as granularity increases, rights delegated to individual sandboxes decrease, limiting the effects of exploited vulnerabilities, in turn forcing attackers to identify and exploit more vulnerabilities to accomplish their goals.

Capability systems are processors, operating systems (OSes), or programming language design to support fine-grained implementation of the principle of least privilege. Each program or component runs with a limited set of rights, represented by its *capabilities*, rather than *ambient authority* as in conventional systems. Capabilities are unforgeable tokens of authority acquired only through new object creation or delegation. The *object-capability* approach blends capabilities with object orientation, linking notions of objects and methods to secure compartments and message passing. *Hybrid capability systems*, such as Cambridge's Capsicum OS model and CHERI processor, are seeing a renaissance with the advent of ubiquitous networking and the resulting need to build more secure and resilient systems.



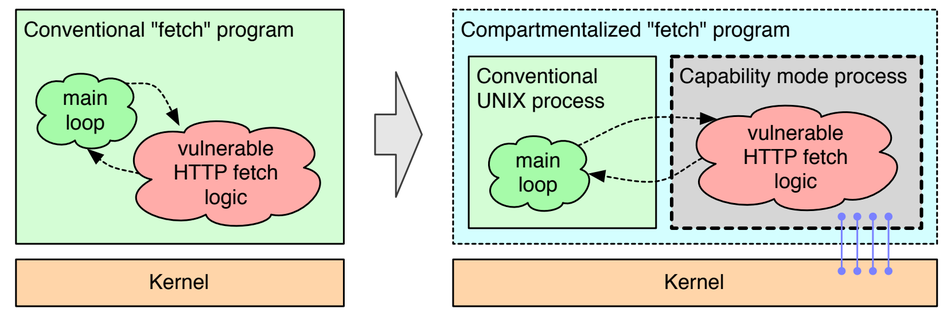
Key aspects of the capability system model were invented at Cambridge during the 1970s, including the CAP Computer (pictured), developed by Wilkes, Needham, and Wheeler, and Karger's SCAP, which implemented capability models in hardware.

Capsicum: practical capabilities for UNIX

Robert N. M. Watson, Jonathan Anderson, Kris Kennaway (Google), Ben Laurie (Google)

To date, *application compartmentalisation*, or *privilege separation*, has been built on weak or semantically mismatched OS foundations, such as chroot and setuid – tools intended to solve different problems. Capsicum is a *hybrid capability model* that blends the *capability system model* with conventional UNIX security in order to provide the benefits of capability security **and** access to mainstream applications. Capsicum adds fine-grained capabilities and a sandboxed *capability mode* to existing APIs. This allows applications to implement their own, often dynamic, policies using capability delegation. A key benefit to Capsicum is that it allows mapping of application security requirements – e.g., the web's *same origin policy* – into robust OS primitives. Capsicum also supports application-level concepts such as Multi-Document Interfaces (MDIs), which map poorly onto MAC systems such as Type Enforcement.

Capsicum's hybrid model allows software authors to reap immediate benefits as they incrementally convert systems to compartmentalisation, and offers a long-term capability system vision inspired by the principle of least privilege. Capsicum shipped in version 9.0 of the widely used open source FreeBSD operating system, and Google has an adaptation of Capsicum to their Linux-based thin-client ChromeOS. Google and the FreeBSD Foundation are sponsoring continued Capsicum development.

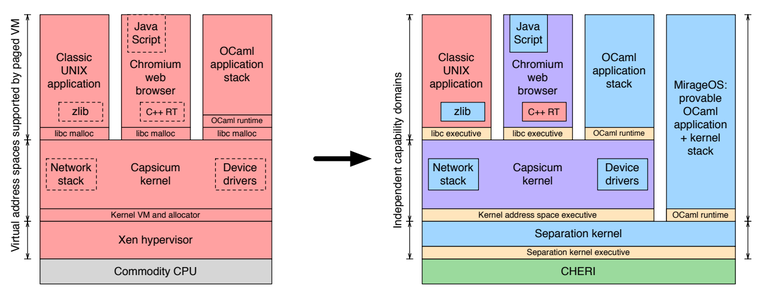


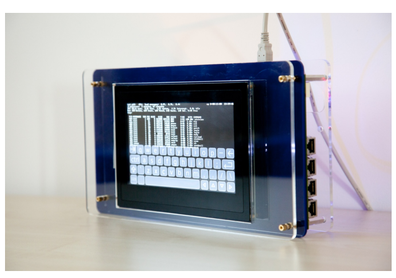
CHERI: a hybrid-capability CPU architecture

Robert N. M. Watson, Peter G. Neuman (SRI), Jonathan Anderson, Ross Anderson, David Chisnall, Nirav Dave (SRI), Brooks Davis (SRI), Khilan Gudka, Steven M. Hand, Ben Laurie (Google), Simon W. Moore, Alan Mujumdar, Steven Murdoch, Robert Norton, Michael Roe, Hassen Saidi (SRI), Jonathan Woodruff, Bjoern Zeeb

Today's CPU instruction set architectures (ISAs) reflect a 1990s consensus on virtual memory. Current systems are exposed to greater threats, placing strain on protection facilities designed for less risky workloads. In the DARPA-funded CTSRD project, SRI International and Cambridge are investigating approaches grounded in the capability security model that will support orders of magnitude greater protection granularity for operating system and application compartmentalisation:

- Capability hardware enhanced RISC instructions (CHERI)
- Apply Capsicum's hybrid capability system model at the ISA level
- De-conflate address space virtualisation and protection
- Support three orders of magnitude greater scaling of protection in low-level TCBs (OS kernels and programming language runtimes)
- Conventional and capability-aware code execute side-by-side
- Incrementally deployable to software TCBs & high-risk components
- Prototyped using 64-bit MIPS ISA; adaptable to 64-bit ARM
- Formal methods applied throughout hardware, ISA, and software
- This approach is now demonstrable using an FPGA-synthesisable soft core to enforce fine-grained protection in a microkernel.





Legacy application code compiled for general-purpose registers

Hybrid code blending general-purpose registers and capabilities

High-assurance capability-only code; stand-alone or "pools of capabilities"

Per-address space memory management and capability executive

SRI International

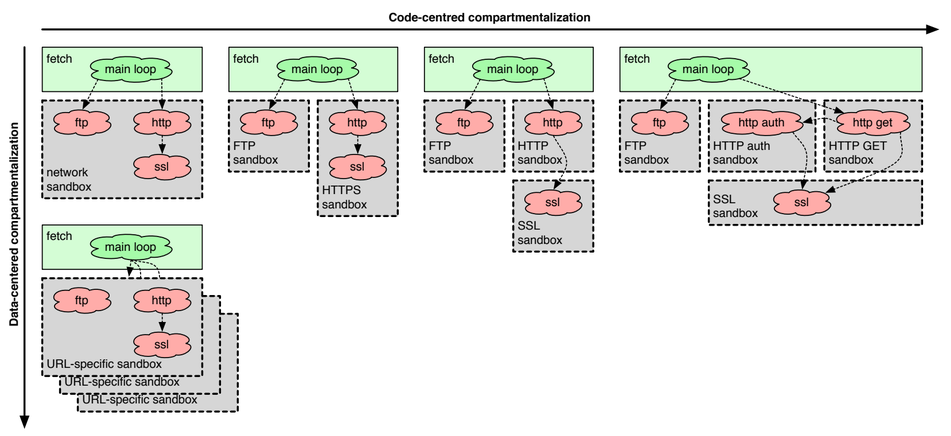
UNIVERSITY OF CAMBRIDGE

Security-oriented analysis of application programs (SOAAP)

Robert N. M. Watson, Khilan Gudka, Steven Hand, Ben Laurie (Google), Anil Madhavapeddy

Application compartmentalisation decomposes software into sandboxes to mitigate security vulnerabilities, and has proven effective in limiting exploits. However, experience shows that adapting existing C programs is difficult, leading to problems with correctness, performance, complexity, and critically, security. Security-Oriented Analysis of Application Programs (SOAAP) is a Google- and DARPA-sponsored project investigating new semi-automated techniques to support programmers in compartmentalisation efforts.

SOAAP allows *compartmentalisation hypotheses* to be explored through source code annotations; it can also be used to find bugs in existing sandboxed applications. Annotations describe which methods to sandbox, which global state and file descriptors sandboxes can access, limits on system services imposed by the sandboxing platform, and what rights are available via RPC. SOAAP uses static and dynamic analysis to engage the developer in an interactive dialogue, identifying potential correctness bugs (e.g., data inconsistencies), and security breaches (e.g., information leaks). Hypotheses are iterative, so programs do not need to be decomposed entirely: a single mitigated vulnerability in a sandbox gives an immediate improvement.



UNIVERSITY OF CAMBRIDGE

Distributed system security (7/8)

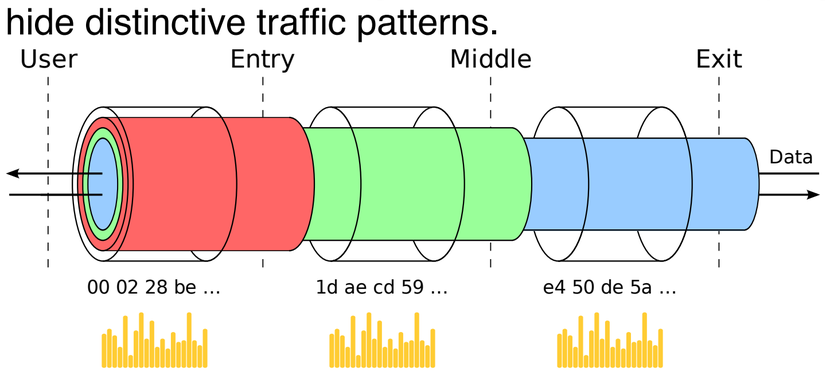
Anonymous communication

Steven Murdoch, Richard Clayton, Robert N.M. Watson

Encryption, as sometimes used with web browsing (SSL/TLS) and email (e.g. PGP), only hides message content, and not the traffic data: source, destination, size and timing. Traffic analysis is the study of such data to discover the behaviour and interests of groups and individuals. It is widely used to track people, for marketing, law-enforcement and by criminals. Anonymity systems, such as Tor, protect the privacy of Internet users from traffic analysis.

Tor is primarily used for anonymous web browsing, and is built from a network of around 1,500 servers (nodes) run by volunteers throughout the world. Messages are encrypted then sent through a randomly chosen path of three servers, within nested layers of encryption. This makes it difficult for an attacker to follow a message between source and destination. In addition to anonymity, Tor is used to circumvent national censorship systems. By hiding what websites a user is accessing, Tor makes it more difficult for countries to block access to certain websites.

However, while Tor prevents message content from being used to match incoming and outgoing connections, it does not significantly distort traffic patterns, allowing traffic analysis to occur. Anonymous communications research at Cambridge, funded by The Tor Project, includes studying the effectiveness of traffic analysis techniques, designing routing methods to reduce the likelihood of traffic being monitored, and developing efficient methods to hide distinctive traffic patterns.



Enforcing user specified policy

Jatinder Singh, Jean Bacon

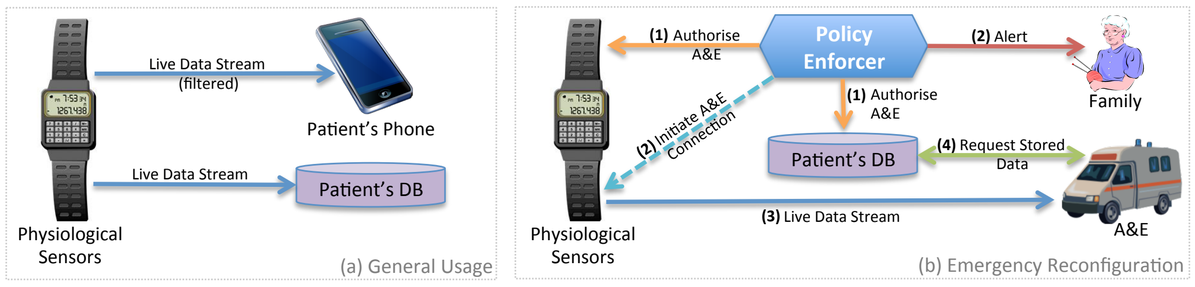
As computing becomes ubiquitous, we notice that:

a) services tend to be personalised, tailored to each particular user
b) services are being provisioned over an ever increasing number of system components, which are often grounded in different administrative domains
c) data may be live (streaming) or stored, control mechanisms must account for both.

Users have preferences as to how, when and with whom their data is shared. Such preferences depend on the circumstances; for instance, a user may choose to relax some restrictions in a medical emergency.

This research concerns the active enforcement of user-specified governance policy in emerging distributed systems. The goal is to enable consistent enforcement across application, system, and administrative boundaries.

Specifically, we consider policy-based middleware, where policy effects user preferences by reconfiguring systems in response to changes in context, triggered by events. This involves issues of policy specification, access control, stream management, component visibility/discovery, data filtering, alerting mechanisms, event composition, and connection management (initiation/cessation/diversion).



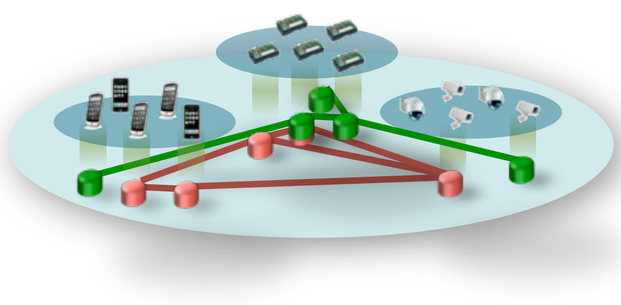
An assisted living environment reconfigured for a medical emergency

FRESNEL: federated secure sensor network laboratory

Jon Crowcroft, Christos Efstratiou, Ilias Leontiadis, Cecilia Mascolo

Current sensor networks

- Single owner
- Single purpose
- Fixed
- One application running
- Difficult to reprogram



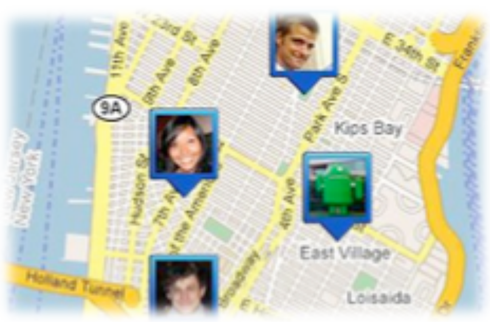
FRESNEL

- Shared sensor networks
- Support multiple applications simultaneously
- Each application can use any sensor available

Privacy and Security

- Different roles have different requirements.
- Sensor network owners have their own security and privacy policies.
- Users and application may have different demands.

FRESNEL aims to offer a framework that supports merging of policies as specified by different roles in the system.

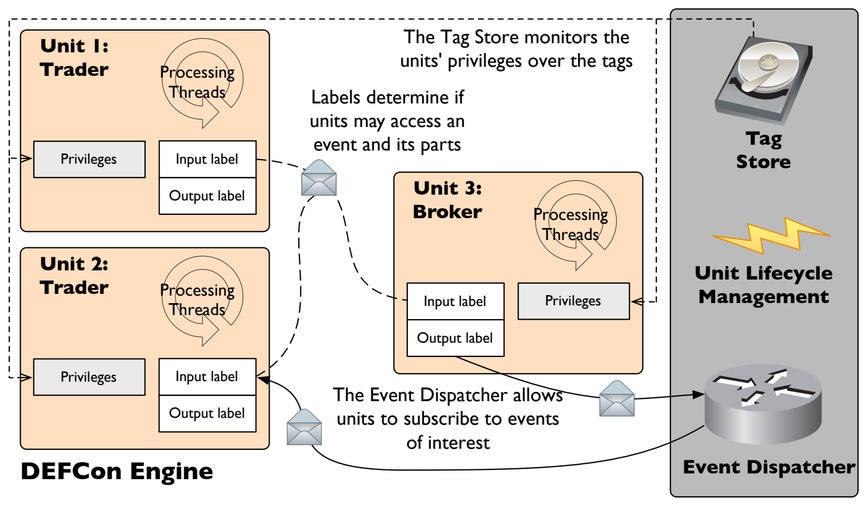


Information flow control in distributed systems: applications to healthcare

David Evans (Derby), Jean Bacon, Jatinder Singh

Healthcare providers world-wide are developing electronic solutions to improve patient care and reduce costs. The resulting systems must not compromise patient safety and privacy. Not surprisingly, healthcare IT efforts in many countries suffer from cost explosion and project overruns. In these systems, middleware software acts as the "plumbing" that integrates many heterogeneous applications, coordinating widely distributed operations.

Decentralised Event Flow Control (DEFCon) is our model for building secure event-based applications. We track sensitive information flows at the granularity of events by using tags to form security labels, expressing confidentiality and integrity requirements; tags are applied to event parts. Privileges over tags constrain how recipients can perceive and propagate events. The figure below shows a DEFCon Engine hosting three processing Units. Every node participating in a DEFCon infrastructure provides a DEFCon Engine. The Engine is the container for event processing within a DEFCon application and includes a trusted kernel. Application code is run by event processing units that are hosted by the Engine. We have an implementation for Java.



UNIVERSITY OF CAMBRIDGE

Cloud computing security (8/8)

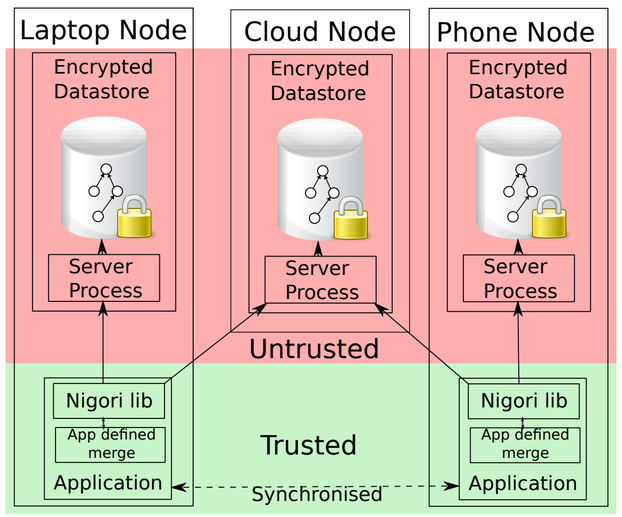
Nigori: Storing secrets in the cloud

Alastair Beresford, Ben Laurie (Google), Andrew Rice, Daniel Thomas

Computer users today have a smartphone, a tablet, a laptop and a desktop machine. Consequently, many new computer applications seamlessly synchronise user data between devices using cloud storage as a highly-available intermediary. Whilst the communication link between the user device and cloud storage is often encrypted, user data is typically stored in a form which is readable by the cloud provider and the application developer.

The aim of the Nigori project is to develop a practical, application neutral, mechanism for storing sensitive user data in the cloud in such a way that the cloud provider and application developer cannot read any of the stored information. We have an initial specification, and an implementation of Nigori for Java and Android. Work will shortly be underway on a JavaScript version suitable for use as a plug-in for Web browsers.

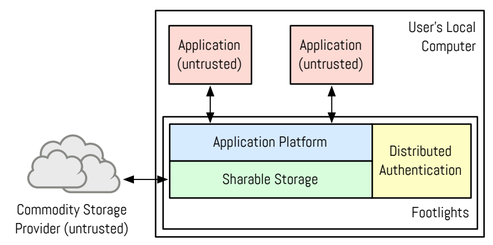
Nigori consists of two components: a datastore and a client library. A Nigori datastore is a service, either run locally on the device alongside the application, or run remotely in the cloud. The client library forms part of the application and runs on a user's device, encrypts data, and manages the user's datastores. A typical application deployment will contain one datastore on each user device and one datastore in the cloud; the application can then use Nigori to keep datastores, and therefore user data, synchronised across all their devices.



Footlights: a secure substrate for social sharing

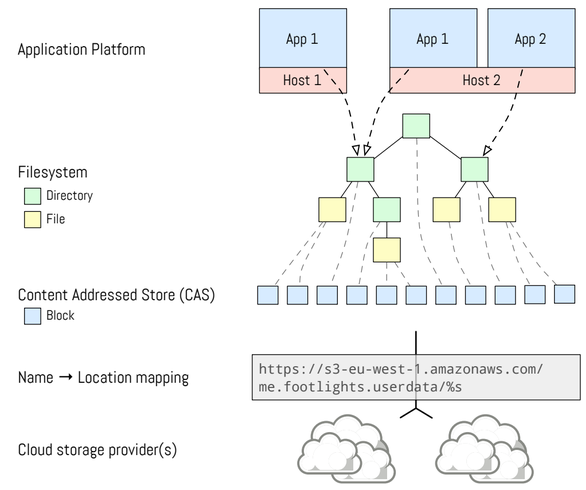
Jonathan Anderson, Frank Stajano

Centralised online social networks (OSNs) are incredibly popular, but have been demonstrated to leak users' private information. Sometimes this leakage is accidental, but other times it is a matter of policy and the business model of today's OSNs such as Facebook.



Footlights explores an alternative model, a hybrid that exploits centralised infrastructure for performance purposes and distributed cryptography techniques for user privacy.

Footlights uses commodity cloud storage as the backing for a distributed encrypted filesystem. Content is broken into fixed-size ciphertext blocks whose linkages are only visible after decryption. Blocks are immutable and deterministically named, so all users can share a global storage namespace without global identities: authorisation is independent of authentication.



Footlights exposes this filesystem to applications that run on the user's computer, where they can be confined and their activity can be observed. Applications can perform arbitrary computation, but access to user data is always anonymous and usually implicit and indirect.

Finally, a distributed authentication scheme allows users to connect to their friends and benefit from social applications any time, anywhere.

MRC²: Resilient, secure data-centre switching

Robert N. M. Watson, Peter G. Neumann (SRI), Simon W. Moore, Andrew Moore, Steven Hand, Anil Madhavapeddy, Alan Mujumdar, Matt Grosvenor, Robert Norton, Phil Porras (SRI), Charalampos Rotsos, Hassen Saidi (SRI), Muhammad Shabhaz, Vinod Yegneswaren (SRI), Dongting Yu, Bjoern Zeeb

This DARPA-sponsored project at SRI and Cambridge is enhancing the data-centre switching security and resilience from several perspectives:

- Resilient Distributed Switch Fabric (RDSF) employs higher-dimensionality links to improve performance, resilience, security, and energy efficiency through greater adaptiveness.
- The CHIMERA memory interconnect offers a weak coherency, capability-oriented memory semantic, improving scalability as the CHERI model scales up to thousands of cores.
- The SCIEL distributed computation framework, and FABLE reconfigurable I/O layer, allow applications to map high-level security and resilience properties into a variety of communications substrates, including RDSF and CHIMERA.
- Trustworthy Programmable Switch Controllers (TPSC) incorporate a formally verified switch implementation with the CHERI CPU in order to provide a secure foundations for distributed switching applications. TPSC also considers distributed switch control through extensions to current software-defined networking (SDN) models.
- Cloud Analysis and Misuse Detection (CAMD) deploys SDN-based security applications in an MRC2 data centre, requiring revisions to mechanism and policy, eliminating global visibility requirements.