Cryptanalysis of the Megamos Crypto Automotive Immobilizer

ROEL VERDULT AND FLAVIO D. GARCIA

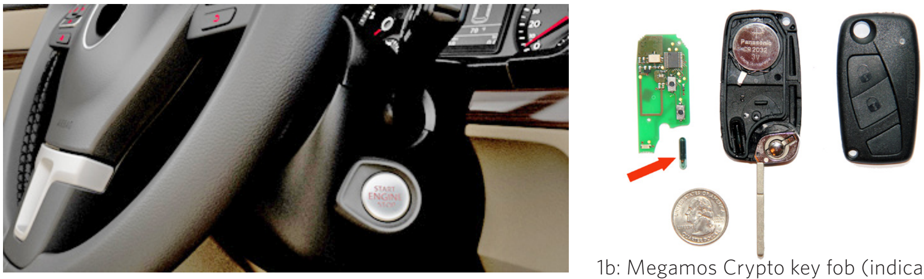
Roel Verdult performed scientific research in a variety of security topics, including electronic passports, contactless smart cards, radio frequency identification (RFID), near field communication (NFC), secure storage, authentication protocols, and other types of transmission security. The relevance of his work is demonstrated by his numerous significant international research awards. He earned his doctorate at two universities, receiving a dual degree from Radboud University, the Netherlands, and KU Leuven, Belgium. Currently, Roel Verdult is active as a cryptographic research engineer and is co-founder of the Dutch IT Security & Engineering company FactorIT BV. roel@factorit.nl

Dr. Flavio Garcia is a Senior Lecturer (Associate Professor) and Senior Birmingham Fellow at the University of Birmingham in the UK. His work focuses on the design and evaluation of cryptographic primitives and protocols for embedded devices like automotive key fobs and smart cards. His research achievements include breakthroughs such as the discovery of vulnerabilities in the four most widely used contactless smart cards: the Mifare Classic, HID iClass, and Atmel's SecureMemory and CryptoRF. The first of these, Mifare Classic, was widely used for electronic payment (e.g., Oyster Card) and access control (e.g., Amsterdam Airport). f.garcia@bham.ac.uk

The Megamos Crypto key fob is used in one of the most widely deployed automotive electronic immobilizers. Such an anti-theft device is designed to prevent hot-wiring of the vehicle. We have reverse-engineered all proprietary security mechanisms of the key fob and have found several weaknesses in the cipher and also in their usage and configuration by carmakers. We exploit these weaknesses in three practical attacks that recover the 96-bit key fob secret key. We end our article with suggestions to mitigate some of our attacks, something that knowledgeable users can do themselves.

Electronic vehicle immobilizers have been very effective at reducing car theft. Such an immobilizer is an electronic device that prevents the engine of the vehicle from starting when the corresponding key fob is not present. This key fob is a low-frequency RFID chip typically embedded in the vehicle's key. When the driver starts the vehicle, the car authenticates the key fob before starting the engine, thus preventing hot-wiring. In newer vehicles the mechanical ignition key has often been removed and replaced by a start button (see Figure 1a). In such vehicles the immobilizer key fob is the only anti-theft mechanism that prevents a hijacker from driving away with the vehicle. In some countries, having such an immobilizer is enforced by law. For example, according to European Commission directive (95/56/EC) it is mandatory that all cars sold in the EU from 1995 on be fitted with an electronic immobilizer. Similar regulations apply to other countries like Australia, New Zealand (AS/NZS 4601:1999), and Canada (CAN/ULC S338-98). Although it is not required by law in the US, according to the independent organization Insurance Institute for Highway Safety (IIHS), 86 percent of all new passenger cars sold in the US had an engine immobilizer installed.

An electronic car immobilizer consists of three main components: a small key fob chip embedded in (the plastic part of) the car key (Figure 1b); an antenna coil located in the dashboard of the vehicle, typically around the ignition barrel; and the immobilizer unit that prevents the vehicle from starting the engine when the key fob is absent.



1a: Keyless ignition with start button

1b: Megamos Crypto key fob (indicated by arrow) in a car key

Figure 1: Megamos Crypto integration in vehicular systems

Make	Models
Alfa Romeo	147, 156, GT
Audi	A1, A2, A3, **A4 (2000)**, A6, **A8 (1998)**, Allroad, Cabrio, Coupé, Q7, S2, S3, S4, S6, S8, **TT (2000)**
Buick	Regal
Cadillac	CTS-V, SRX
Chevrolet	Aveo, Kalos, Matiz, Nubira, Spark, Evanda, Tacuma
Citroën	**Jumper (2008)**, Relay
Daewoo	Kalos, Lanos, Leganza, Matiz, Nubira, Tacuma
DAF	CF, LF, XF
Ferrari	California, 612 Schaglietti
Fiat	Albea, Doblo, Idea, Mille, Multipla, Palio, **Punto (2002)**, Seicento, Siena, **Stilo (2001), Ducato (2004)**
Holden	Barina, Frontera
Honda	Accord, Civic, CR-V, FR-V, HR-V, Insight, **Jazz (2002, 2006)**, Legend, Logo, S2000, Shuttle, Stream
Isuzu	Rodeo
Iveco	Eurocargo, Daily
Kia	Carnival, Clarus, Pride, Shuma, Sportage
Lancia	Lybra, Musa, Thesis, Ypsilon
Maserati	Quattroporte
Opel	Frontera
Pontiac	G3
Porsche	911, 968, Boxster
Seat	Altea, Cordoba, **Ibiza (2014)**, Leon, Toledo
Skoda	**Fabia (2011)**, Felicia, Octavia, Roomster, Super, Yeti
Ssangyong	Korando, Musso, Rexton
Tagaz	Road Partner
Volkswagen	Amarok, Beetle, Bora, Caddy, Crafter, Cross Golf, Dasher, Eos, Fox, Gol, **Golf (2006, 2008)**, Individual, Jetta, Multivan, New Beetle, Parati, Polo, Quantum, Rabbit, Saveiro, Santana, **Scirocco (2011)**, Touran, **Tiguan (2010)**, Voyage, **Passat (1998, 2005)**, Transporter
Volvo	C30, **S40 (2005)**, S60, S80, **V50 (2005)**, V70, XC70, XC90, XC94

Table 1: Vehicles that used Megamos Crypto for some version/year. Boldface and year indicate specific vehicles we experimented with.

The immobilizer unit communicates through the antenna coil and enumerates all key fobs that are in proximity of the field. The key fob identifies itself and waits for further instructions. The immobilizer challenges the key fob and authenticates itself first. On a successful authentication of the immobilizer unit, the key fob sends back its own cryptographic response, which is different every time. Only when this response is correct does the immobilizer unit enable the engine to start.

The immobilizer unit is directly connected to the internal board computer of the car, also referred to as the electronic control unit (ECU). To prevent hot-wiring a car, the ECU blocks fuel-injection, disables spark plugs, and deactivates the ignition circuit if the key fob fails to authenticate.

A distinction needs to be made between the vehicle immobilizer and the remotely operated central locking system. The latter is battery powered, operates at ultra-high frequency (UHF), and only activates when the user pushes a button on the remote to (un)lock the doors of the vehicle. Figure 1b shows a disassembled car key where it is possible to see the passive Megamos Crypto key fob and also the battery powered remote of the central locking system.

The Megamos Crypto key fob is the first cryptographic immobilizer key fob manufactured by EM Microelectronic-Marin SA and is currently one of the most widely used. The manufacturer claims to have sold more than 100 million immobilizer chips, including Megamos Crypto key fobs [4]. Table 1 shows a list of vehicles that use or have used Megamos Crypto at least for some version/year. As can be seen from this list, many Audi, Fiat, Honda, Volkswagen, and Volvo cars used Megamos Crypto key fobs.

The key fob uses a 96-bit secret key and a proprietary cipher in order to authenticate to the vehicle. Furthermore, a 32-bit pin code is needed in order to be able to write on the memory of the key fob. The concrete details regarding the cipher design and authentication protocol are kept secret by the manufacturer, and little is currently known about them.

From our collaboration with the local police it was made clear to us that sometimes cars are being stolen and nobody can explain how. They strongly suspect the use of so-called “car diagnostic” devices. Such a device uses all kinds of custom and proprietary techniques to bypass the immobilizer and start a car without a genuine key. This motivated us to evaluate the security of vehicle immobilizer key fobs.

In the last decades, semiconductor companies introduced several proprietary algorithms specifically for immobilizer security. Their security often depends on the secrecy of the algorithm, contrary to Kerckhoffs’ principle. When their inner-workings are uncovered, it is often only a matter of weeks before the first attack is published. There are several examples in the literature that address the insecurity of proprietary algorithms [5]. There are four widely used immobilizer key fobs that depend on proprietary cryptography: DST, KeeLoq, Hitag2, and Megamos Crypto, which were all proven to be insecure [1, 2, 7, 8]. The Megamos paper was accepted at the 22nd USENIX Security Symposium, but appears as an addendum to the 24th’s Proceedings, and is used as a basis for this article.

Hardware Setup

We used a Proxmark III (http://www.proxmark.org/) to eavesdrop and communicate with the car and key fob. This is a generic RFID protocol analysis tool that supports raw data sampling of radio frequency signals [6]. We have developed a generic open

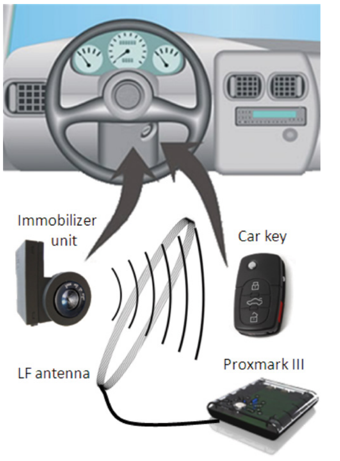


Figure 2: Experimental setup for eavesdropping

source library, which is capable of supporting any custom and proprietary RFID communication scheme that operates at a frequency of 125 kHz. This allowed us to implement a custom firmware and FPGA design that uses the modulation and encoding schemes of Megamos Crypto key fobs.

Furthermore, we added RFID reader/programmer functionality to send simple commands like read and write to the key fob. In particular, this library can be used to set the memory lock bit and a random pin code as a mitigation for our second attack, as described later in this article. Finally, we implemented an advanced firmware, which contains all cryptographic operations and is fully compatible with the Megamos Crypto authentication protocol. This enabled us to perform practical experiments with cars by eavesdropping and emulation of Megamos Crypto key fobs. However, we will not release any attack tools such as this advanced firmware.

Megamos Crypto

This section gives a short introduction to the workings of the Megamos Crypto key fob. It briefly introduces the cryptographic algorithms and protocols used in Megamos Crypto; a more detailed description is available in [8].

Authentication Protocol

The car authenticates by sending a random nonce n_C and the corresponding car authenticator a_C. When the car successfully authenticates itself, the Megamos Crypto key fob responds with its own key fob authenticator a_T back to the car. A simplified version of the Megamos Crypto authentication protocol is shown in Figure 3.

When the driver turns on the ignition, several messages between the car and key fob are exchanged. It starts with the car reading out the key fob memory blocks. Next, the car tries to authenticate using the shared secret key k. If the authentication fails, the car retries around 20 times before it reports on the dashboard that the immobilizer failed to authenticate the key fob. Table 2 shows an eavesdropped trace of a German car that initializes and authenticates a Megamos Crypto key fob using the 96-bit key 000000000000010405050905. The structure of the secret key of the car suggests that it has an entropy of only 24 bits.

Cryptographic Algorithm

Several after-market diagnostic and locksmith tools such as the *Tmpro2, MiraClone, AVDI,* and *Tango Programmer* implement the Megamos Crypto cipher for key fob production and verification. None of these tools is able to recover the secret key of a key fob or perform any kind of cryptanalysis. However, the software package that comes with Tango Programmer implements all cryptographic operations of the key fob, including the Megamos Crypto cipher. We have analyzed the software thoroughly and extracted the algorithm from it. The Megamos Crypto cipher is a stream cipher that consists of five main components: a 23-bit Galois Linear Feedback Shift Register, a 13-bit Non-Linear Feedback Shift Register, and three 7-bit registers.

The stream cipher basically works as a pseudo-random generator that is seeded by the secret key k and the car nonce n_C. It then runs producing pseudo-random bit-strings a_C and a_T, which are used in the authentication protocol as proof of knowledge of the secret key (see Figure 4).

Cryptanalysis of Megamos Crypto

In our full paper [8], we have proposed a cryptanalysis that compromises *all* vehicles using Megamos Crypto. This cryptanalysis requires an adversary to eavesdrop two successful authentication traces between the car and the key fob to recover the 96-bit secret key. We would like to emphasize that in order to get these two traces, a perpetrator needs access to both the car and the original car key. Our cryptanalysis reduces the computational complexity from 2^{96} (a brute force attack) to 2^{56} encryptions.

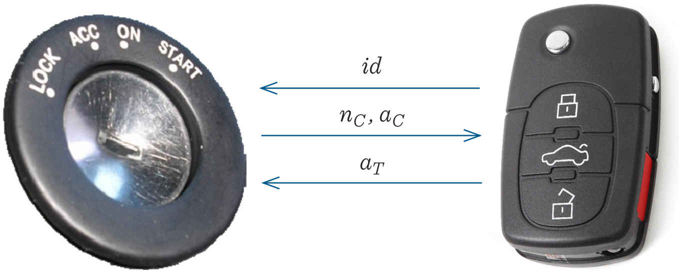


Figure 3: Megamos Crypto authentication protocol

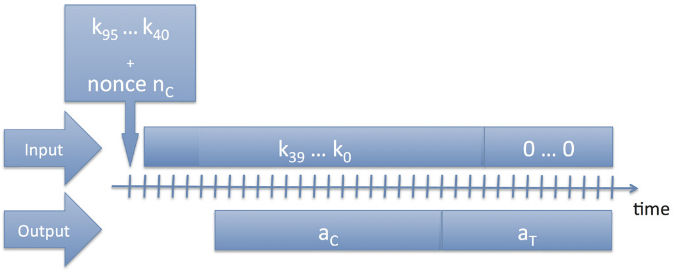


Figure 4: Initialization and propagation of the cipher

Origin	Message	Description
Car	3	Read identifier
Key fob	A9 08 4D EC	Identifier $id_{31} \dots id_0$
Car	6 | 3F FE 1F B6 CC 51 3F | 0^7 | F3 55 F1 A	Authentication, $n_{C_{55}} \dots \mathrm{n}_{C_0}, 0^7, \mathrm{a}_C$
Key fob	60 9D 6	Car authenticated successfully, send back a_T

Table 2: Eavesdropped Megamos Crypto authentication trace

This could be computed within two days on a COPACOBANA, which is an FPGA-based massively parallel computer.

Once the secret key is recovered, it is possible to emulate the original key fob, effectively cloning the original key. The cryptanalysis described above exploits the following weaknesses.

- The key fob lacks a pseudo-random number generator, which makes the authentication protocol vulnerable to replay attacks.
- The internal state of the cipher consists of only 56 bits, which is much smaller than the 96-bit secret key.
- The cipher state successor function can be inverted; given an internal state and the corresponding bit of cipher-text, it is possible to compute the predecessor state.
- The last steps of the authentication protocol provide an adversary with 15-bits of known-plaintext.

This computational complexity can be further reduced by a time/memory tradeoff. Many tradeoffs are possible, but if we were to use a 12-terabyte lookup table, for example, then the complexity is reduced to 2^{49} table lookups. This optimized version of the attack takes advantage of the fact that some of the cipher components can be run quite autonomously. Such a time-memory tradeoff, however, requires many indirect memory lookups and is therefore difficult to mount in practice with ordinary consumer hardware.

Partial Key-Update Attack

Currently, the memory of many Megamos Crypto key fobs in the field is either unlocked or locked with a publicly known default pin code. This means that anybody has write access to the memory of the key fob. This also holds for the secret key bits that make it vulnerable to a trivial denial of service attack. An adversary just needs to flip one bit of the secret key of the key fob to disable it.

Besides this obvious weakness, there is another weakness regarding the way in which the 96-bit secret key is written to the key fob. These 96 bits are stored in six memory blocks of 16 bits each. But it is only possible to write one block at a time to the key fob, which constitutes a serious weakness since a secure key-update must be an atomic operation.

This weakness enables an adversary to use a guess-and-determine technique in which she overwrites one block of the key at a time until she finds the complete secret key. For this attack we assume that an adversary is able to communicate with the car and key fob. A slightly optimized version of this attack requires only one successful authentication trace. In total, we need to write three times on the memory of the key fob and perform 3 × 2^{16} authentications with the key fob. This can be done within 30 minutes using a Proxmark III. The computational complexity of the last three steps is 2^{15} encryptions, which takes less than a second on an ordinary laptop.

We have executed this attack in practice and recovered the secret key of several cars from various makes and models. Having recovered the key, we were able to emulate the key fob and start the vehicles.

Weak-Key Attack

Our third attack is based on the following observation: many of the keys that we have recovered using the previous attack had very low entropy and exhibited a well-defined pattern, i.e., the first 32 bits of the key were all zeros. In the remainder of this paper we call such a key *weak*. This attack consists of a time-memory tradeoff that exploits this weakness to recover the secret key, within a few minutes, from two authentication traces. This attack requires storage of a 1.5 TB rainbow table.

Table 3 shows some examples of weak keys we found during our experiments (on the vehicles indicated in Table 1). To avoid naming concrete car models we use *A, B, C*...to represent car makes. We write numbers *X*.1, *X*.2, *X*.3...to represent different car models of make *X*.

Car	Secret key
A.1	00000000d8 b3967c5a3c3b29
A.2	00000000d9 b79d7a5b3c3b28
B.1	0000000000 00010405050905

Table 3: Recovered keys from our own cars. Besides the evident 32 leading zero bits, every second nibble seems to encode a manufacturer-dependent value, which further reduces the entropy of the key.

Apparently, some car manufacturers have decided to use only 64 bits of the secret key, probably due to compatibility issues with legacy immobilizer systems. If a Megamos Crypto key fob uses such a weak key, it is possible to recover this key quickly, even when the memory of the key fob is locked with a pin code. Concretely, if the first 32 bits of the key are constant (e.g., zeros), this allows an adversary to pre-compute and sort on 47 contiguous output bits for each internal state. However, such a table, with 2^{56} entries, requires a huge amount of storage. Many time-memory tradeoff methods have been proposed in the literature. For example, a rainbow table shrinks the storage significantly, while requiring only a modest amount of computation for a lookup. Just to give an impression of the feasibility of this attack, if we were using a rainbow table of 1.5 TB, then the computational complexity required to perform this attack would only be 2^{37} encryptions, which can be computed within a few minutes on a standard laptop.

Practical Considerations and Mitigation

Our attacks require close-range wireless communication with both the immobilizer unit and the key fob. It is not hard to imagine real-life situations, like valet parking or car rental, where an adversary has access to both for a period of time. It is also possible to foresee a setup with two perpetrators, one interacting with the car and one wirelessly pickpocketing the car key from the victim's pocket.

As mitigating measures, car manufacturers should set uniformly generated secret keys and, for the devices which are not locked yet, set pin codes and writelock their memory after initialization. These obvious measures would prevent a denial of service attack, our partial key-update attack described earlier, and our weak-key attack in the previous section.

Car owners can protect their own vehicles against a denial of service and the partial key-update attack. These attacks only work if the adversary has write access to the memory of the key fob, which means that the lock-bit is set to zero. It is possible for a user to test for this property with any compatible RFID reader, like the Proxmark III, using our communication library. If the lock-bit is set to zero, then you should set it to one. It is possible to set this bit without knowing the secret key or the pin code. When dealing with the more recent version of the Megamos Crypto key fob (EM4170), users should also update the pin code to a random bit-string before locking the key fob.

On the positive side, our first (cryptographic) attack is more computationally intensive than the other attacks, which makes it important to take the aforementioned mitigating measures in order to prevent the more inexpensive attacks. Unfortunately, our first attack is also hard to mitigate when the adversary has access to the car and the key fob (e.g., valet parking or car rental). It seems infeasible to prevent an adversary from gathering two authentication traces. Furthermore, this attack exploits weaknesses in the core of the cipher's design (e.g., the size of the internal state). It would require a complete redesign of the cipher to fix these weaknesses. To that purpose, lightweight ciphers like Grain, Present, and KATAN have been proposed in the literature and could be considered as suitable replacements for Megamos Crypto. Also, immobilizer products implementing AES are currently available in the market.

Conclusions

The implications of the attacks presented in this paper are especially serious for those vehicles with keyless ignition. At some point the mechanical key was removed from the vehicle, but the cryptographic mechanisms were not strengthened to compensate. We want to emphasize that it is important for the automotive industry to migrate from weak proprietary ciphers like this to community-reviewed ciphers such as AES and use them according to the guidelines. For a few years already, there have been contactless smart cards on the market that implement AES and have a fairly good pseudo-random number generator. It is surprising that the automotive industry is reluctant to migrate to such key fobs considering the cost difference of a better chip (≤ 1 USD) in relation to the prices of high-end car models (≥ 50,000 USD). Since most car keys are actually fairly big, the key fob design does not really have to comply with the (legacy) constraints of minimal size.

Following the principle of responsible disclosure, we notified the manufacturer of our findings back in November 2012. Since then we have maintained an open communication channel with them. We understand that measures have been taken to prevent the weak-key and partial key-update attacks when the key fob was improperly configured.

Acknowledgments

The authors would like to thank Ross Anderson, Tom Chothia, Riccardo Focardi, Dorine Gebbink, Casey Henderson, Bart Jacobs, Sam King, Bas Kortmann, Karsten Nohl, Carolyn Pike, Jon Rowe, Mark Ryan, Graham Steel for their firm support. We are also thankful to Robert Carolina and Kenny Paterson for getting involved [3].

References

[1] A. Bogdanov, "Linear Slide Attacks on the Keeloq Block Cipher," in *Information Security and Cryptology* (2008), vol. 4990 of *Lecture Notes in Computer Science*, Springer, pp. 66–80.

[2] S. C. Bono, M. Green, A. Stubblefield, A. Juels, A. D. Rubin, and M. Szydlo, "Security Analysis of a Cryptographically-Enabled RFID Device," in *Proceedings of the 14th USENIX Security Symposium (USENIX Security 2005)* (2005), USENIX Association, pp. 1–16.

[3] R. Carolina, and K. G. Paterson, "Megamos Crypto, Responsible Disclosure, and the Chilling Effect of Volkswagen Aktiengesellschaft vs. Garcia et al.": http://www.isg.rhul.ac.uk/~kp/ Carolina-Paterson-Megamos-comment-20130828.pdf.

[4] 125 kHz crypto read/write contactless identification device, EM4170, product datasheet, March 2002, EM Microelectronic-Marin SA.

[5] R. Verdult, "The (In)security of Proprietary Cryptography," PhD thesis, Radboud University, The Netherlands, and KU Leuven, Belgium, April 2015.

[6] R. Verdult, G. de Koning Gans, and F. D. Garcia, "A Toolbox for RFID Protocol Analysis," in *Proceedings of the 4th International EURASIP Workshop on RFID Technology (EURASIP RFID 2012)* (2012), IEEE Computer Society, pp. 27–34.

[7] R. Verdult, F. D. Garcia, , and J. Balasch, "Gone in 360 Seconds: Hijacking with Hitag2," in *Proceedings of the 21st USENIX Security Symposium (USENIX Security 2012)* (2012), USENIX Association, pp. 237–252.

[8] R. Verdult, F. D. Garcia, and B. Ege, "Dismantling Megamos Crypto: Wirelessly Lockpicking a Vehicle Immobilizer," in *Supplement to the 22nd USENIX Security Symposium (USENIX Security 2013)* (2015), USENIX Association, pp. 703–718.

**Do you have a USENIX Representative on your university or college campus?
If not, USENIX is interested in having one!**

The USENIX Campus Rep Program is a network of representatives at campuses around the world who provide Association information to students, and encourage student involvement in USENIX. This is a volunteer program, for which USENIX is always looking for academics to participate. The program is designed for faculty or staff who directly interact with students. We fund one representative from a campus at a time.

A campus rep's responsibilities include:

- Maintaining a library (online and in print) of USENIX publications at your university for student use
- Distributing calls for papers and upcoming event brochures, and re-distributing informational emails from USENIX
- Encouraging students to apply for travel grants to conferences
- Providing students who wish to join USENIX with information and applications
- Helping students to submit research papers to relevant USENIX conferences
- Providing USENIX with feedback and suggestions on how the organization can better serve students

In return for being our "eyes and ears" on campus, the Campus Representative receives access to the members-only areas of the USENIX Web site, free conference registration once a year (after one full year of service as a Campus Representative), and electronic conference proceedings for downloading onto your campus server so that all students, staff, and faculty have access.

To qualify as a campus representative, you must:

- Be full-time faculty or staff at a four-year accredited university
- Have been a dues-paying member of USENIX for at least one full year in the past

**For more information about our Student Programs, contact
Julie Miller, Marketing Communications Manager, julie@usenix.org**

usenix
THE ADVANCED
COMPUTING SYSTEMS
ASSOCIATION