By JOHANNES BUCHMANN, ALEXANDER MAY, and ULRICH VOLLMER

PERSPECTIVES FOR CRYPTOGRAPHIC LONG-TERM SECURITY

Cryptographic long-term security is needed, but difficult to achieve. Use flexible cryptographic tools, and have replacements ready.

Many activities, including communication, commerce, medical care, public administration, politics, and education depend heavily on information technology. IT systems of the future will be highly dynamic. Mobile networks will consist of billions of heterogeneously computing devices: sensors, RFIDs, PDAs, personal computers, teraflop servers, perhaps even quantum computers. IT infrastructures are vulnerable and their vulnerability will increase. Protection of IT infrastructures is a very complex task and is becoming one of the main challenges of computer science research. Cryptography serves as a foundation for most IT security solutions. Three illustrative examples include:

- Digital signatures are used to prove the authenticity of automatic software updates, for example for operating systems such as Microsoft Windows XP. The update is digitally signed by the issuer. Only if the digital signature is valid, the update is installed. It is absolutely crucial that this authentication works correctly. If an attacker is able to inject malicious software into an update for Microsoft Windows XP, then, given the huge market share of Microsoft, this could potentially cause a catastrophe.
- Home banking, e-commerce, and many other Web applications use the Secure Socket Layer protocol (SSL) or Transport Layer Security (TLS) for authentication and data encryption. SSL and TLS use digital signatures for authentication and the Diffie-Hellman protocol for the exchange of secret encryption keys. There were more than 220,000 SSL/TLS secured servers worldwide circa November 2005.

Illustration by Richard Downs

- Germany is beginning to introduce an electronic health card for its citizens in 2006 (see [6]). The health card will allow access to medical insurance data, prescriptions, and medication records including a drug usage history. For this purpose, the health card uses digital signatures and encryption schemes.

But how long will the cryptography that we use today remain secure? Is this time long enough to keep private information such as medical data confidential? Do we have adequate alternatives to current cryptographic techniques? What shall we do if a cryptographic algorithm turns out to be insecure?

How Secure is Today's Cryptography?

First, we consider signatures. In their landmark *Communications* article in 1978, Rivest, Shamir, and Adleman [9] proposed the famous RSA scheme that is still the most important digital signature algorithm. For example, it is used by Microsoft to digitally sign its operating system updates. It is also used in SSL/TLS, S/Mime, and in the German electronic health card. The security of RSA relies on the intractability of factoring composite integers, the so-called RSA moduli, which are generated as the product of two large prime numbers. In their original article, the inventors of RSA suggested using an RSA modulus having 200 decimal digits for long-term security. Later, the company RSA Security published a list of RSA moduli of increasing sizes and offered an award for factoring those numbers (see www.rsasecurity.com). In 1993, the first 120-digit RSA challenge was factored. In 2005, Bahr et al. announced the successful factorization of a 200-digit number from the challenges. So in the past 30 years there has been considerable progress in factoring RSA moduli. This progress is due to ingenious algorithmic ideas such as Pollard's number field sieve as well as advances in computer and implementation technology. Based on theory and experiments, Lenstra and Verheul [7] have developed an interpolation formula that predicts the future security for RSA and other relevant cryptographic functions (see www.keylength.com). According to this formula, RSA remains secure for the next 30 years if an RSA modulus of at least 880 digits is chosen.

Brilliant mathematical ideas that make the integer factorization problem feasible are always possible. For example, in 1996 Shor [10] showed that the construction of quantum computers—a new type of computing device that makes use of the laws of quantum mechanics—would make factoring RSA moduli feasible and would therefore break RSA. However, today it is still unclear whether quantum computers with sufficiently large registers can ever be built. The largest existing quantum computer is only able to factor the number 15.

There are a number of alternatives to the RSA signature algorithm: for example, the Digital Signature Algorithm DSA. Its security is based on the so-called discrete logarithm problem (DLP) in the multiplicative group of a finite field. The status of the DLP in finite fields is similar to the status of the integer factoring problem. There has been considerable algorithmic and technological progress in the past 30 years. The current record was announced in 2005 by Joux and Lercier, who succeeded in finding discrete logarithms in a prime field with 130-digit characteristic. Analogous to the factorization problem, the construction of sufficiently large quantum computers would make DLP in finite fields feasible. For the time being, however, the DLP in finite fields is intractable, if appropriate parameters are chosen.

Another important alternative to RSA is elliptic curve cryptography (ECC). Its security is based on the DLP in the group of points of an elliptic curve over a finite field. The elliptic curve discrete logarithm problem currently seems to be much more difficult than either the integer factorization problem or the DLP in finite fields. Algorithmic progress in this area has been slow. Therefore, much smaller keys can be chosen in ECC than in RSA and DSA. This makes ECC well suited for small computing devices. For example, digital signatures on the German electronic passport are ECC signatures. Again, sufficiently large quantum computers would break ECC.

We now consider hash functions, which are used as electronic fingerprints of data, particularly in digital signature schemes. In signature schemes, a potentially long document is first hashed to a short bit string, the message's fingerprint. Instead of signing the document itself, only the short fingerprint is signed. Assume an attacker is able to find a hash function collision, that is, two different documents $d \neq d'$ having the same fingerprint. In this case, the attacker can replace document d by d', since both have identical fingerprints and therefore identical signatures. Therefore, we require that cryptographic hash functions provide collision intractability. Contemporary cryptographers consider a hash function as intractable if more than 2^{80} hash values are required in order to find a collision.

Unfortunately, current hash functions seem to

To prepare for the future and unexpected attacks two things are necessary. A pool of secure alternative cryptographic algorithms must be made available and the applications that use cryptography must be designed in a modular way such that insecure primitives can be easily replaced by secure ones.

have an even shorter lifetime than encryption schemes. In 1990, Rivest proposed the MD4 hash algorithm. Only six years later, Dobbertin showed that collisions in MD4 can be found by computing no more than 2^{20} hash values. Recently, Wang, Lai, Feng, and Yu showed that only 2^{8} hash values suffice to find a collision with probability at least 2^{-6}. MD4 was followed by MD5 in 1992. Recent results of Wang and Yu show that the complexity for finding a collision in MD5 is only 2^{39}. For the NIST standard SHA-1 the complexity of the best attacks also beats the presumed 2^{80} hash requests. SHA-1 became a standard in 1995. Wang, Yin, and Yu recently showed an attack on SHA-1 within complexity 2^{69}.

Although the use of SHA-1 might still provide enough security for most applications today, the cryptographic community must put considerable efforts into the search for better design criteria for the long-term security of hash functions.

How about the long-term security of encryption? In 1977, the Data Encryption Standard (DES) was approved as a Federal standard. Twenty years later in 1997, the DESCHALL project announced breaking a message encrypted with DES. Another year later, the Electronic Frontier Foundation developed special hardware called Deep Crack that broke a DES key in 56 hours. Even though the normal DES algorithm was designed very carefully, it could not provide confidentiality of a document for more than 20 years.

Is Today's Cryptography Good Enough?

We have seen there is a growing need for IT security in general and cryptographic tools to support this security in particular. Some of the applications only require short-term security. Examples are code signing and strong authentication in SSL. Other applications require long-term security, for example encryption of sensitive medical data or digital signatures for contracts. Is the cryptography that we have today appropriate and sufficiently secure for these purposes?

Our experience with cryptosystems indicates that carefully designed cryptographic primitives have an expected lifetime of 5 to 20 years. So presently, using cryptographic primitives such as RSA, ECC, or AES, the successor of DES, is adequate to achieve short-term security. But what do we do 20 years from now? Also, how do we react when unexpected progress in cryptanalysis makes cryptographic primitives obsolete much earlier? To prepare for the future and unexpected attacks two things are necessary. A pool of secure alternative cryptographic algorithms must be made available and the applications that use cryptography must be designed in a modular way such that insecure primitives can be easily replaced by secure ones.

The efficient replacement of insecure cryptography requires that all applications import all their cryptography from a dedicated crypto API such as the Java Cryptographic Architecture (JCA) or the Microsoft Crypto API. In addition, protocols are needed that execute the replacement of cryptography. Such protocols must not only replace the crypto primitives. They also have to replace keys, certificates, and so forth. Following these criteria, we have designed and implemented the crypto library Flexi-Provider [5] that supports the JCA and implements mainstream and alternative cryptography.

We have also designed and implemented the trust center software FlexiTrust [1] that imports its cryptography from our FlexiProvider. FlexiTrust is being used by the German National Root Certification Authority (CA) and the German Country Signing CA. Moreover, we implemented a plug-in that enables Microsoft Outlook to use any signature algorithm that is implemented in the FlexiProvider. So it seems that cryptographic flexibility can be achieved with present technology.

To maintain IT security in the future, we need to work toward a rich portfolio of viable cryptosystems. This portfolio must include adequate primitives for the ubiquitous computing context, and systems that remain secure in the presence of large quantum computers.

What about alternatives for currently used cryptographic algorithms such as RSA, DSA, and ECC? Since in the near future quantum computers may become a serious threat to cryptography, research and development of new cryptosystems should focus on those alternatives that have a chance of remaining secure even in the presence of quantum computers. We describe a few of the most promising candidates, see also [3]. Experimental versions of these candidates are implemented in our PostQuantumProvider, which is part of the FlexiProvider.

The security of the NTRU encryption scheme presumes the difficulty of computing a shortest vector in a lattice of large dimension (SVP). In moderately large dimensions, solving SVP is far from being feasible, even with new algorithmic ideas such as Schnorr's random sampling [2]. Quantum algorithms currently provide only a minor speed-up in solving SVP.

The McEliece and Niederreiter encryption schemes rely on the difficulty of the decoding problem for certain classes of error-correcting codes, in particular permuted Goppa codes. At the moment, the best way to solve the decoding problem is to transform it into the so-called Low-Weight-Code-Word-Problem (LWCP). But finding solutions of LWCP in large dimensions seems to be infeasible.

The security of the signature scheme SFlash is based on the difficulty of solving (underdetermined) systems of multivariate quadratic equations over finite fields (MVQS). The standard method for solving MVQS involves computing the Gröbner basis of the system. Although the algorithms might take advantage of the special structure of the equations in MVQS-cryptosystems, it seems infeasible to solve systems with many hundreds of equations. As is the case for the decoding problem, no quantum algorithms are known that significantly enhance or improve on the speed of classical algorithms.

The Merkle signature scheme [4] allows for an a priori fixed number of signatures to be made with a given key. It is very efficient, and its security relies only on the one-wayness and collision resistance of the used hash function.

There are in fact many potential alternative cryptographic schemes that have a chance of providing security, even in the presence of large quantum computers. However, a lot of additional research is necessary. Each of the new systems requires careful analysis in both the classical as well as the quantum model, analysis that is still scant outside the well-trodden paths of RSA. Once sufficient confidence in a system and its security parameters is established, standardization must guarantee interoperability well before the rollout in a flexible infrastructure that guarantees modularity.

The world of ubiquitous computing generates even more new requirements for cryptographic primitives. Current digital signature and encryption schemes cannot easily be used in small devices such as RFIDs or small embedded systems since they require too much computing power and storage. Thus, cryptographic research will also have to provide adequate primitives for the ubiquitous computing context.

What about cryptographic long-term security? According to German laws authentic medical data must remain accessible for at least 30 years. For example, in case of medical malpractice, it may be necessary to reconstruct the course of medical treatment. This reconstruction might require the verification of very old digital signatures. Today's digital signatures, however, do not guarantee the desired long-term security.

To remedy this, Maseberg has suggested using multiple signatures [8]. Suppose a document bears two digital signatures. Provided the two signature

schemes used are sufficiently independent, one of the two signature schemes is likely to remain secure, even if the other is broken. The insecure signature scheme can be replaced by a new secure one. Afterward the document has to be re-signed. Again we have two valid signatures of our document. In this way, the authenticity of our document can in principle remain assured an infinite time. Maseberg has proposed protocols that extend standard protocols of a public-key infrastructure and support multiple signatures including the update management in the case of a break, all with little overhead.

Long-term confidentiality is much more difficult to achieve. If a document is multiple encrypted, then it is only confidential until the strongest of the encryption schemes becomes insecure. In contrast to digital signatures, re-encryption makes no sense: An attacker can store the original ciphertext and then decrypt this ciphertext as soon as the strongest encryption scheme is broken. For long-term confidentiality it is therefore necessary to use extremely strong cryptosystems with sufficiently large keys. These keys must be agreed upon between sender and recipient of confidential data.

Quantum cryptography (QC) solves the problem of long-term secure encryption at least partially. QC provides primitives for key agreements whose security relies on the laws of quantum mechanics and information theory and not on the difficulty of certain computational problems. Those primitives are secure forever unless quantum mechanics turns out to be incorrect. Its success notwithstanding, QC is no panacea. QC is not yet sufficiently efficient and many basic cryptographic primitives cannot be realized within the QC framework. The search for long-term secure cryptography continues.

Conclusion

Today's cryptography provides strong tools for short-term security. Applications can rely on those tools as long as they are flexible enough to quickly replace cryptographic components that become insecure—which can always happen since cryptanalytic progress is difficult to predict. To maintain IT security in the future, we need to work toward a rich portfolio of viable cryptosystems. This portfolio must include adequate primitives for the ubiquitous computing context, and systems that remain secure in the presence of large quantum computers. We have shown there are many promising candidates, each of which requires careful analysis both in the classical as well as in the quantum model, implementation, and standardization. One of our contributions to this process is the open source library FlexiProvider. We have shown how long-term security for digital signatures can be achieved by means of multiple signatures. Long-term confidentiality of encrypted data has turned out to be the most challenging open problem. A potential partial solution might come from the development of quantum cryptography.

References

1. Buchmann, J. et al. An evaluated certification services system for the German national root CA—Legally binding and trustworthy transactions in e-business and e-government. In *Proceedings of The 2005 International Conference on e-Business, Enterprise Information Systems, e-Government, and Outsourcing (EEE'05)*, 103–108.
2. Buchmann, J. and Ludwig, C. Practical lattice basis sampling reduction. To appear in *Proceedings of the 7th Algorithmic Number Theory Symposium (ANTS-VII)*.
3. Buchmann, J. et al. Post quantum signatures; eprint.iacr.org/2004/297.pdf.
4. Coronado, C. Provably secure and practical signature schemes. Doctoral Thesis, Technische Universität Darmstadt, 2005; elib.tu-darmstadt.de/diss/000642
5. *FlexiProvider—A Toolkit for the Java Cryptography Architecture (JCA/JCE)*; www.flexiprovider.de.
6. German Federal Ministry of Health and Social Security. Information on the electronic health card. www.die-gesundheitsreform.de/presse/publikationen/pdf/flyer_elektronische_gesundheitskarte_en.pdf.
7. Lenstra, A.K. and Verheul, E.R. Selecting cryptographic key sizes. *Journal of Cryptology: The Journal of the International Association for Cryptologic Research 14*, 4 (2001), 255–293.
8. Maseberg, J.S. Fail-safe konzept für public-key infrastrukturen. Doctoral Thesis, Technische Universität Darmstadt, 2002.
9. Rivest, R., Shamir, A., and Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM 21*, 2 Feb. 1978), 120–126.
10. Shor, P.W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. In S. Goldwasser, Ed., *Proceedings of the 35th Annual Symposium on foundations of Computer Science (FOCS 1994)*, 124–134

Johannes Buchmann (buchmann@cdc.informatik.tu-darmstadt.de) is a professor in the Departments of Computer Science and Mathematics and vice president of research of the Technische Universität Darmstadt in Germany.

Alexander May (may@cdc.informatik.tu-darmstadt.de) is a junior professor in the Department of Computer Science at the Technische Universität Darmstadt in Germany.

Ulrich Vollmer (uvollmer@cdc.informatik.tu-darmstadt.de) is a post doctoral member of the Department of Computer Science at the Technische Universität Darmstadt in Germany.