From the Academy. In the article "Security and privacy in the information economy" by Joan Feigenbaum, Steven Rudich, Matt Blaze, and Kevin McCurley, which appeared in number 7, April 1, 1997, of *Proc. Natl. Acad. Sci. USA* (**94,** 2789–2792), the authors wish to point out a printer's error. The annotations to the bibliographic material, which were omitted on the page proof, were reinserted incorrectly prior to publication. The full text with correct annotations is shown below.

1. Luby, M. (1996) *Pseudorandomness and Cryptographic Applications*, (Princeton Univ. Press, Princeton, NJ).
   (Highly mathematically rigorous treatment of much of the theory of security and privacy. Accessible only to specialists in the theory of computation. Excellent as a reference book or as a textbook for an advanced course.)
2. Diffie W. & Hellman, M. (1976) *IEEE Transactions on Information Theory*, Vol. IT-22, pp. 644–654.
   (Seminal research paper that establishes the foundations of public-key cryptography. Accessible primarily to computer scientists but possibly to computer scientifically aware people in other fields.)
3. Merkle, R. (1978) *Commun. Assoc. Computing Machinery* **21,** 294–299.
   (Seminal research paper that establishes the foundations of public-key cryptography. Accessible primarily to computer scientists but possibly to computer scientifically aware people in other fields.)
4. Brassard, G., (1988) *Modern Cryptography: A Tutorial, Lecture Notes in Computer Science*, (Springer, Berlin) Vol. 325.
   (Dated but well written introduction to the theory of security and privacy, accessible to computer scientifically aware nonspecialists.)
5. Simmons, G., ed. (1992) *Contemporary Cryptology: The Science of Information Integrity*, (IEEE Press, New York).
   (Collection of survey papers on many aspects of cryptology and its applications. Long introduction, some of which is suitable for nonspecialists, some only for specialists.)
6. Stinson, D. (1995) *Cryptography: Theory and Practice*, (CRC Press, Boca Raton, FL).
   (Introductory textbook intended for advanced undergraduate or beginning graduate courses.)
7. Landau, S. (1983) *Notices Am. Math. Soc.* **30,** 7–10.
   (Article about the Rivest–Shamir–Adleman (RSA) public-key cryptosystem and the controversy that surrounded its invention. Accessible to scientifically educated nonspecialists.)
8. Landau, S. (1988) *Notices Am. Math. Soc.* **35,** 5–12.
   (Introduction to the notion of "zero-knowledge proof system," put forth by Goldwasser, Micali, and Rackoff, in which a "prover" and "verifier" exchange messages; the prover convinces the verifier that a string *x* is in a set *S*, and the verifier "learns," in a precise technical sense, nothing but this one bit of information. The article also discusses the practically important variant called "zero-knowledge proofs of identity," put forth by Fiat and Shamir, and the controversy that surrounded its invention. Accessible to scientifically educated nonspecialists.)
9. Shor, P. (1994) *Proceedings of the 35th Symposium on Foundations of Computer Science*, (IEEE Computer Soc. Press, Los Alamitos, CA) pp. 124–134.
   (Breakthrough research paper, accessible only to specialists in the theory of computation.)
10. Bennett, C., Brassard, G. & Ekert, A. (1992) *Sci. Am.* **266,** 50–57.
    (Survey article accessible to scientifically educated nonspecialists.)

Biochemistry. In the article "BAP-135, a target for Bruton's tyrosine kinase in response to B cell receptor engagement" by Weiyi Yang and Stephen Desiderio, which appeared in number 2, January 21, 1997, of *Proc. Natl. Acad. Sci. USA* (**94,** 604–609), due to a printer's error, the communicated line was omitted from the manuscript. This line should read: *Communicated by Max D. Cooper, University of Alabama at Birmingham, Birmingham, AL, November 5, 1996 (received for review September 23, 1996*).

Biochemistry. In the article "Enhancer blocking activity located near the 3′ end of the sea urchin early H2A histone gene" by Franco Palla, Raffaella Melfi, Letizia Anello, Maria Di Bernardo, and Giovanni Spinelli, which appeared in number 6, March 18, 1997, of *Proc. Natl. Acad. Sci. USA* (**94,** 2272–2277), the following printer's error should be noted. On page 2276, the last sentence of the first paragraph should read: "From these results we may exclude that *sns* represses transcription by an active silencing mechanism with the binding of one or more repressor molecules." In the printed paper, the word "exclude" was incorrectly replaced with the word "suggest."

Proc. Natl. Acad. Sci. USA
Vol. 94, pp. 2789–2792, April 1997
From the Academy

This paper serves as a summary of a symposium session as part of the Frontiers of Science series, held November 7–9, 1996, at the Arnold and Mabel Beckman Center of the National Academies of Sciences & Engineering in Irvine, CA.

Security and privacy in the information economy

JOAN FEIGENBAUM*, STEVEN RUDICH†, MATT BLAZE*, AND KEVIN MCCURLEY‡

*AT&T Laboratories, Murray Hill, NJ 07974-0636; †Carnegie Mellon University, Computer Science Department, Pittsburgh, PA 15213-3891; and ‡Sandia National Laboratory, Albuquerque, NM 87185-0129

Relentless progress in computing and communications technology has brought the world to a point at which many millions of people have routine access to powerful computers and networks. Consequently, more and more of the world's information is created, used, transmitted, and stored electronically. These advances give rise to major concerns about the authenticity, integrity, and privacy of our information. They also create opportunities for mathematicians, computer scientists, and computing and communication technologists to define, develop, and deploy the techniques needed to enhance our security and privacy while equipping us with faster, more affordable, and more efficient means of conducting our daily business. In short, after many years of hype, the "information economy" is finally becoming a reality, and it has brought security and privacy research to center stage in the scientific community.

In this session, we approach the question of "security and privacy in the information economy" as computer science researchers. There are two major components to the computer science research point of view: namely that of the theorist, whose goal is to develop a rigorous mathematical theory of security and privacy, and that of the practitioner, whose goal is to build systems that are secure and private and that are widely and successfully used in the real world. After many years of working fairly independently, security theorists and security practitioners are now cooperating to meet the pressing demands of the information economy.

Steven Rudich, an Associate Professor of Computer Science at Carnegie Mellon University in Pittsburgh, gave the first talk of the session, an overview of security and privacy theory. The fundamental conjecture of this theory is that "one-way functions" exist. Informally, a function is one-way if it is easy to compute but hard to invert. Completely mathematically rigorous definitions of "easy" and "hard" are well established. For purposes of this session, it suffices to interpret these terms as follows: A function f is one-way if one can compute $f(x)$ in a reasonable amount of time on any input x in the domain of f, but one cannot, given y for which there is an x such that $f(x) = y$, compute such an x in a reasonable amount of time. Note that the existence of one-way functions is the "fundamental conjecture" of security and privacy theory, not the "fundamental theorem." There is, at the moment, no known proof that such functions exist, and proving their existence is intimately related to the notorious "P vs. NP" problem that lies at the heart of almost all important open questions in the theory of computation. Nonetheless, there are well understood functions in elementary mathematics that are believed to be one-way and that have remained uninvertible in practice despite centuries of search by experts for efficient methods to invert them. Here are two functions widely believed to be one-way:

Multiplication of Primes. Here the domain of f is the set of pairs (P, Q), where P and Q are primes, and the value of f at (P, Q) is simply the product $N = PQ$. Inverting f thus consists of factoring integers known to be the product of two primes. Standard multiplication programs running on today's computers can easily multiply two 250-digit primes. Yet, the fastest known factoring program cannot, given a 500-digit N known to be the product of two primes, find these primes within 10^{36} years.

Discrete Exponentiation. In this example, the domain of f is the set of triples (p, g, e), where p is a prime, g is a generator of the multiplicative group modulo p, and e is an integer in the range $[0, p - 2]$. The value of f at (p, g, e) is the triple (p, g, x), where x is congruent to g^e modulo p. To invert f, one must take a triple (p, g, x) and find the unique e such that x is congruent to g^e modulo p, which is called "the discrete logarithm of x with respect to (g, p)." As in the previous example, routinely available programs can compute f quickly when p, g, and e are each 500 digits, but no program has ever been devised that can invert f on 500-digit numbers within 10^{36} years.

It should be noted that even "hard" problems may be easy to solve on particular inputs. For example, if the input to the multiplication function is of the form $3Q$ for a prime Q, then the algorithm that simply checks for divisibility by 3 will quickly discover this and be able to invert these cases. The theory of one-way functions addresses these deficiencies in a way that is beyond the scope of this short article.

Interesting special cases of one-way functions are the "trapdoor functions," so-called because inverting such a function is made feasible by the possession of some auxiliary information, the "trapdoor." (Converting this informal description to a technically correct definition is subtle, and interested readers should consult one of the references below.) These functions are at the heart of a crucial enabling technology for the information economy: public key cryptography. Classical cryptosystems consist of a pair of functions E and D (for "encrypt" and "decrypt") and work as follows. Two parties, say Alice and Bob, agree on a shared random string k, which they use as a "key" in the encryption scheme. When Alice wants to send a private message x to Bob, she computes $y = E(x, k)$ and sends y over a public channel (e.g., a path in today's Internet). Bob receives y and computes $x = D(y, k)$. An eavesdropper, because he does not know k, cannot recover x (or any part of it) from y. Such systems cannot do the entire job of providing privacy in the information economy because of the need for Alice and Bob to share a secret key before they can conduct business; if Bob is an Internet merchant, for example, he does not want to require potential customers to obtain encryption keys that will be valid only for his business before they can send him orders privately. In a public key cryptosystem, each user has a pair of keys: a public key K_1 and a private key K_2. If Alice wants to send x privately to Bob, she obtains Bob's public K_1, computes $y = E(x, K_1)$, and sends y to Bob. When he receives y, Bob uses his private K_2 to compute $x = D(y, K_2)$. Anyone can obtain Bob's public key and send him encrypted messages, but the only

0027-8424/97/942789-4$2.00/0
PNAS is available online at **http://www.pnas.org**.

known way to decrypt these messages is to use the corresponding private key, which is known only to Bob. Thus, the function determined by the encryption method E and the public key K_1 is a trapdoor function. An eavesdropper cannot invert this function to recover the secret message x, but Bob, armed with the trapdoor information K_2, can invert it easily.

The ingenious notion of associating a (public key, private key) pair with each user rather than a single key with each pair of users, also facilitates "digital signature schemes," which are vitally important for Internet commerce. In such a scheme, there are two functions, S for "sign" and V for "verify." To sign a digital document w in the course of a transaction, Bob computes $z = S(w, K_2)$ and makes the pair (w, z) part of the record of the transaction. Anyone needing to verify that it was indeed Bob who computed z looks up the public key K_1 associated with Bob and runs the verification procedure $V(w, z, K_1)$; V should return "ACCEPT" if $z = S(w, K_2)$ and otherwise return "REJECT." Note that digital signatures (necessarily) differ from physical signatures in an important way. The signature z is a function of the document w as well as the signer Bob. Here are two examples of public key schemes, one with stronger provable properties than the other.

In the RSA public key cryptosystem (named for its inventors Rivest, Shamir, and Adleman), a user (say Bob) chooses his keys as follows. He first generates two random primes P and Q that are large enough so that their product $N = PQ$ cannot be factored by any known method within an acceptable amount of time. He then finds an integer e in the range $[3, (P - 1)(Q - 1)]$ with no factors in common with $P - 1$ or $Q - 1$, and he finds the unique corresponding integer d such that the product ed is congruent to 1 modulo $(P - 1)(Q - 1)$. (Such quadruples P, Q, e, d are plentiful and can be found using well understood, standard software.) His public key is (N, e), and his private key is (P, Q, d). (In fact, he does not need to save P and Q after he has generated N, d, and e, but he does need to keep them private, i.e., to ensure that no one else gets them.) To send a message to Bob, Alice must first break it into "blocks," each of which is represented as an integer in the range $[0, N - 1]$. The encryption $E(x, N, e)$ of block x is $y = x^e$ modulo N. To decrypt y, Bob computes $D(y, N, d) = y^d$ modulo N. Elementary facts about modular arithmetic suffice to show that, for all x, $D(E(x, N, e), N, d) = x$—that is, if one first encrypts and then decrypts, one gets back the block x that one started with. The trapdoor function in this system is exponentiation modulo N, where N is the product of two primes. The trapdoor information is the factorization of N (or, equivalently, the decryption exponent d). It is crucial that all arithmetic in the RSA system be done modulo N—ordinary integer arithmetic would not work. The reason for this is that root-finding is easy over the integers. Given an integer y that is the e^{th} power of some integer x, one can use a standard procedure to find x. In arithmetic modulo N, where N is the product of primes P and Q, no such procedure is known. The use of modular arithmetic also ensures that a ciphertext block y is of the same length as a plain text block x; if arithmetic were done over the integers, then y would be e times as long as x, which is obviously undesirable for large exponents e.

Notice that, because the two functions in the RSA scheme commute, they can be used for signature as well as encryption. To sign block w, Bob simply computes $z = D(w, N, d)$; to verify that z is indeed Bob's signature on w, Alice looks up Bob's public key (N, e) and checks that $w = E(z, N, e)$. The only known way to "break" the RSA system (i.e., to be able, in general, to forge Bob's signature or decrypt a ciphertext block sent to Bob if all you know is Bob's public key) is to factor the modulus N—that is, to find the trapdoor information. Because multiplication of primes is believed to be a one-way function, it is not widely believed that this attack on the system would work. However, breaking the system has not (yet?) been rigorously proven to be equivalent to factoring the modulus. The possibility remains that another break will be discovered; users must weigh this possibility against the fact that the system has been studied intensely since it was first published in 1978, and factoring the modulus is still the only known way to break it.

In the Blum–Goldwasser public key cryptosystem, Bob generates his private key by choosing two large random primes P and Q each congruent to 3 modulo 4. His public key is simply the product N of P and Q. Messages are represented in binary notation as "bit strings." An important operation is the exclusive-or of two bits, which is just the sum of those bits modulo 2. To send a t-bit message x to Bob, Alice proceeds as follows. First, she chooses an integer r uniformly at random from the integers in $[1, N - 1]$ that are divisible neither by P nor by Q. She then "stretches" r into t "pseudorandom" bits $b_1, \ldots, b_t$ by constructing a sequence of numbers $r_1, \ldots, r_t$. To get b_1, she computes r_1 by squaring r modulo N and takes b_1 to be the least significant bit of r_1; in general, r_i is obtained by squaring $r_{i\text{-}1}$ modulo N, and b_i is the least significant bit of r_i. She sends to Bob the bit sequence $y_1, \ldots, y_t$ (where y_i is the exclusive-or of x_i and b_i) and the number s (which is the square of r_t modulo N). Because he knows the factors P and Q, Bob can compute square roots modulo N; thus he can recover from s first the sequence $r_t, \ldots, r_1$ (r_t is just the square root of s modulo N, and, in general, r_i is the square root of r_{i+1} modulo N), then the bits $b_1, \ldots, b_t$, and finally the original message bits $x_1, \ldots, x_t$. (Note that ex-or'ing with b_i is a "self-inverse" operation; x_i is just the ex-or of y_i and b_i.) Furthermore, it can be rigorously proven that there is no way to "break" the Blum–Goldwasser scheme without factoring the modulus N; the ability to compute x (or even any "meaningful information" about x, in a sense that can be made mathematically precise), given the public key N, the cipher text $y = y_1, \ldots, y_t$, and the number s, is provably equivalent to the ability to factor N. See ref. 2 for a more in-depth (but still accessible) explanation of the Blum–Goldwasser scheme and why breaking it is equivalent to factoring.

Rompel's theorem (1) is one of the glorious achievements of security and privacy theory. If there is a one-way function, then there is a digital signature scheme that is secure against "chosen message attack"—even if an adversary can force his victim to sign a set of messages of the adversary's choosing, he cannot subsequently forge the victim's signature on an arbitrary message not in the already-signed set. In particular, if multiplication of primes is one-way, one can use this fact to build a digital signature scheme that is secure against chosen message attack.

Matt Blaze, a Principal Member of Research Staff at AT&T Labs in New Jersey, gave the second talk of the session and explained the practitioner's point of view. The crux of this viewpoint is that there are serious, although not necessarily insurmountable, obstacles to implementing and using many cryptographic schemes in the computing and communication environment that we have today, even schemes that are perfectly satisfactory according to all important criteria in the theoretical literature. Here are four examples of such obstacles that were covered in the talk or the discussion following:

Practical cryptographic systems are made up of a collection of security components rather than a single cryptographic algorithm with well understood security properties. The precise manner in which cryptographic algorithms are used together (the "security protocol") has as much impact on the security of the system as do the individual algorithms themselves. Unfortunately, many "obviously" secure security protocols turn out to have serious weaknesses that are only discovered well after the systems that implement them have been deployed. Compounding the problem is that implementing security software is a difficult and subtle process that is very easy to do wrong; small changes in the environment can often invalidate the assumptions on which the security of the underlying algorithms are based. For example, keys are usually assumed to be random and unpredictable. Yet the key generation schemes used in several commercial software packages have turned out to have flaws that allowed attackers to easily guess the keys used.

All cryptographic systems assume a secure, private environment in which computations are performed. Today's networked workstations and personal computers, however, do not really provide such an environment. So-called "smart cards," which provide portable security environments, offer a partial solution to this problem, but they are not yet part of the "standard" infrastructure available to most potential users of electronic commerce.

Our standard infrastructure also lacks a reliable distribution mechanism for public keys. Any product or service that needs to receive signed messages from or send encrypted messages to an entity, say X, with which it has had no prior communication must obtain a reliable copy of X's public key. In today's Internet, there are no "phone books of public keys," i.e., highly available, highly accurate, highly tamper-resistant mappings between entities' names and their keys. Nor is it clear what the best notion or notions of "entity" are for this purpose: Should each person be given a (public key, private key) pair at birth that is designed to last a lifetime, or would such a key pair be objectionable for the same reasons that "national identity cards" are objectionable in the physical world? Should each person have many key pairs, one for each "role" that he plays in the information economy? Or should keys not be associated with people at all, but rather with "authorizations" (e.g., to sign purchase orders within spending limits or to co-sign hiring and firing decisions for employees at certain levels) that are conferred upon people in such a way that the names or "identities" of these people are irrelevant or even secret?

Non-technical pressures have limited the spread of cryptography even in applications for which the technical issues are solved. Historically, there has been little demand for cryptography in the commercial world; there has been even less demand when the extra security comes at the expense of reduced performance or increased cost. Furthermore, Cold War era United States Government export regulations treat cryptography as a "munition." Rules and procedures that purport to restrict overseas sale and use of cryptography-based products and services have the effect of restricting the availability of cryptography domestically. Technology companies often cannot cost-effectively develop two versions of their products simultaneously, one for the domestic market and one for export, nor can they truthfully claim to have bested their foreign competitors if they are not free to use state-of-the-art cryptography while their offshore competitors are.

The fundamental notion of one-way function was proposed 20 years ago in a seminal paper of Diffie and Hellman (2) and, independently, in a paper of Merkle (3) that was written concurrently but did not appear until 2 years later. In the past 20 years, the "Science of Modern Cryptography" referred to in the title of the first talk in this session has blossomed abundantly, and there are now published schemes with provable properties that achieve a wide variety of security and privacy goals that could be useful or even necessary for the information economy. (See refs. 1 and 4–6 for a technical introduction to some of these ideas and refs. 7 and 8 for a less technical introduction.) The upcoming years should see the development and deployment of some of those schemes in the real world, the rejection of others as not quite as useful in practice as they seemed in theory, and the creation of new theory that captures more precisely the security and privacy goals that users really have.

The session ended with a lively and wide-ranging discussion. The audience asked questions that prompted clarification and fleshing out of some of the points made during the talks and raised many new points, both technical and nontechnical. Here are two examples of good questions and answers from the discussion.

Question 1: Aren't many of the cryptographic schemes discussed in this session breakable by quantum computers?

In theory, yes. A breakthrough paper by Shor (9) shows that, in the quantum model of computation, neither discrete exponentiation nor integer multiplication of primes is a one-way function. However, there are two reasons that Shor's breakthrough does not spell the imminent end of cryptography as we know it. First, quantum computers, while well defined in theory, do not yet exist in practice, and none of the scientists and engineers now seriously attempting to build them expect practical quantum computers to be widely (if at all) available in the near- to medium-term future (if ever). Second, recall that cryptography, like any theoretical development in computer science, assumes a well formulated, underlying model of computation. Implicit in both talks presented in this session was the Turing machine model, a very useful and accurate one for the computers and networks that exist today. The questioner is completely right that, if one changes one's underlying model of computation, then one's theory of cryptography has to change as well. Fortunately, there is a mature theory of "quantum cryptography," containing many cryptographic schemes that are efficient and unbreakable, in a sense made precise by this theory, in the quantum model of computation and communication (10). Thus, if quantum computers and networks become widely available, we can still have security and privacy; we will simply have to use quantum techniques to achieve them.

Question 2: The talks in this session addressed mostly technical issues. Aren't some of the toughest questions facing us in the "information economy" actually nontechnical? How much of the business of society, now conducted with pens and paper and other things that most people understand, do we really want to migrate to high-speed networks and digital money and other things that most people don't understand? How much of our personal information do we want to deposit in computers and networks, where it will undoubtedly be used in unforeseen ways, some of which we would not permit if we could foresee them?

The question struck a responsive chord, both with the audience and with the presenters of the session. Indeed, social reluctance to accept cryptographic protection of digitized information as an adequate substitute for physical protection of paper-based information is probably one realistic answer to the question in the title of the second talk of the session ("If Cryptography is so Great, Why Isn't Everybody Using It?"). Current highly visible, highly capitalized experiments with cryptography-based Internet commerce present a vast opportunity for purveyors of cryptography and other security and privacy technology to convince the general public that this reluctance should give way to the convenience and efficiency that such technology offers. This is truly an experimental period in the history of cryptography, because the ultimate outcomes of these experiments are far from clear. There is certainly some evidence that people will eventually accept digitized information protected by cryptography as a fact of everyday life, whether they understand how it works or not. It is not as though we now rely solely on paper, pens, and other things that we fully understand to conduct our daily business. How many of us, after all, fully understand how the telephone system works?

1. Luby, M. (1996) *Pseudorandomness and Cryptographic Applications*, (Princeton Univ. Press, Princeton, NJ).
(Survey article accessible to scientifically educated nonspecialists.)
2. Diffie W. & Hellman, M. (1976) *IEEE Transactions on Information Theory*, Vol. IT-22, pp. 644–654.
(Dated but well written introduction to the theory of security and privacy, accessible to computer scientifically aware nonspecialists.)
3. Merkle, R. (1978) *Commun. Assoc. Computing Machinery* **21,** 294–299.
(Seminal research paper that establishes the foundations of public key cryptography. Accessible primarily to computer scientists but possibly to computer scientifically aware people in other fields.)
4. Brassard, G., (1988) *Modern Cryptography: A Tutorial, Lecture Notes in Computer Science*, (Springer, Berlin) Vol. 325.
(Article about the Rivest–Shamir–Adleman (RSA) public-key cryptosystem and the controversy that surrounded its invention. Accessible to scientifically educated nonspecialists.)
5. Simmons, G., ed. (1992) *Contemporary Cryptology: The Science of Information Integrity*, (IEEE Press, New York).
(Introduction to the notion of "zero-knowledge proof system," put forth by Goldwasser, Micali, and Rackoff, in which a "prover" and "verifier" exchange messages; the prover convinces the verifier that a string x is in a set S, and the verifier "learns," in a precise technical sense, nothing but this one bit of information. The article also discusses the practically important variant called "zero-knowledge proofs of identity," put forth by Fiat and Shamir, and the controversy that surrounded its invention. Accessible to scientifically educated nonspecialists.)
6. Stinson, D. (1995) *Cryptography: Theory and Practice*, (CRC Press, Boca Raton, FL).
(Highly mathematically rigorous treatment of much of the theory of security and privacy. Accessible only to specialists in the theory of computation. Excellent as a reference book or as a textbook for an advanced course.)
7. Landau, S. (1983) *Notices Am. Math. Soc.* **30,** 7–10.
(Seminal research paper that establishes the foundations of public key cryptography. Accessible primarily to computer scientists but possibly to computer scientifically aware people in other fields.)
8. Landau, S. (1988) *Notices Am. Math. Soc.* **35,** 5–12.
(Breakthrough research paper, accessible only to specialists in theory of computation.)
9. Shor, P. (1994) *Proceedings of the 35th Symposium on Foundations of Computer Science*, (IEEE Computer Soc. Press, Los Alamitos, CA) pp. 124–134.
(Collection of survey papers on many aspects of cryptology and its applications. Long introduction, some of which is suitable for nonspecialists, some only for specialists.)
10. Bennett, C., Brassard, G. & Ekert, A. (1992) *Sci. Am.* **266,** 50–57.
(Introductory textbook intended for advanced undergraduate or beginning graduate courses.)