Secure and Private Distribution of Online Video and Some Related Cryptographic Issues

Feng Bao, Robert Deng, Peirong Feng, Yan Guo, Hongjun Wu

Kent Ridge Digital Labs
21 Heng Mui Keng Terrace, Singpore 119613
{baofeng, deng, pfeng, yguo, hongjun}@krdl.org.sg

Abstract. With the rapid growth of broadband infrastructure, it is thought that the bottleneck for video-on-demand service through Internet is being cleared. However, digital video content protection and consumers privacy protection emerge as new major obstacles. In this paper we propose an online video distribution system with strong content security and privacy protection. We mainly focus on the study of security and privacy problems related to the system. Besides presenting the new system, we intensively discuss some relevant cryptographic issues, such as content protection, private information retrieval, super-speed encryption/decryption for video, and PKC with fast decryption etc. The paper can be viewed as one that proposes practical solutions to real life problems, as well as one that presents applied cryptography research.

1 Introduction

Television has been elected as one of the greatest inventions in the last century. Public demand on video-based communication, entertainment and education has been the driving force for many technologies, such as broadband network and video compression. Nowadays, people are no longer satisfied with the fixed TV programs. They want to watch what they love to watch, and pay for that, i.e. personalized video service like the services provided in restaurants. To meet this need, Video-on-Demand (VoD) has been studied for many years. [Minoli] is a good reference for the academic and industrial effort for VoD technologies. Researchers have been focusing on how to stream the video to an online Internet consumer without dropping of critical frames. SMIL is ironed out to serve as a standard for synchronized integration of multimedia streams by W3C. It is claimed in [Jai99] that industries have even moved far ahead of academies in this field to step into the new frontier.

With the rapid growth of broadband infrastructure, it is thought that the bottleneck for video-on-demand service through Internet is cleared. Digital content security emerges as a new challenge. Up to now, online video consumers (OVCs) have very limited choice of online video contents, as video content providers (VCPs) hesitate to put their contents in digital format in the network. VCPs are not comfortable with the level of content security provided by the current technology [GMDS98]. On the other hand, online consumers also concern about their privacy being disclosed.

In this paper, we propose an online video distribution scheme that protects VCP s video contents and the consumer s privacy simultaneously. The content protection in

V. Varadharajan and Y. Mu (Eds.): ACISP 2001, LNCS 2119, pp. 190-205, 2001.

our scheme is based on the public key cryptography implemented in a tamper-resistant hardware, which is not a new idea. But we focus more on security discussion and analysis. We also study some cryptographic issues arising from the scheme. The privacy protection is based on a simple PIR(private information retrieval) scheme.

The organization of the paper is as follows. In Section 2 we describe our online video distribution system. In Section 3 we discuss the advantages of using public key cryptography in tamper-resistant hardware for content protection. In Section 4 we study the privacy protection issue in our online video system. The system features are displayed in Section 5. In Section 6, we propose a general method to construct the symmetric key ciphers that have super-speed for video encryption. In Section 7 we propose a public key cryptosystem with fast decryption, which is motivated by implementing decryption in a hardware device with cheap processors. We present the design and analysis of two concrete super-high speed ciphers for video encryption in the Appendix, which can be excluded from the paper.

2 System Description

2.1 Outline of the System

In our online video system there are four parties.

VCP---Video Content Provider,
OVW---Online Video Warehouse,
OVC---Online Video Consumer,
THM---Tamper-resistant Hardware Manufacture

An OVW is an online storage service provider that may support several VCPs. A VCP encrypts its different videos by different secret keys and puts the encrypted videos at an OVW. An OVC can freely download the encrypted videos in his/her favor. The OVC can only watch the video after he/she pays the VCP for the secret key to decrypt the video.

However, the secret key should not be given to OVC plainly for content protection. That key should be given to OVC in the encrypted form and be decrypted in a tamper-resistant hardware device (produce by THM) as described in the following.

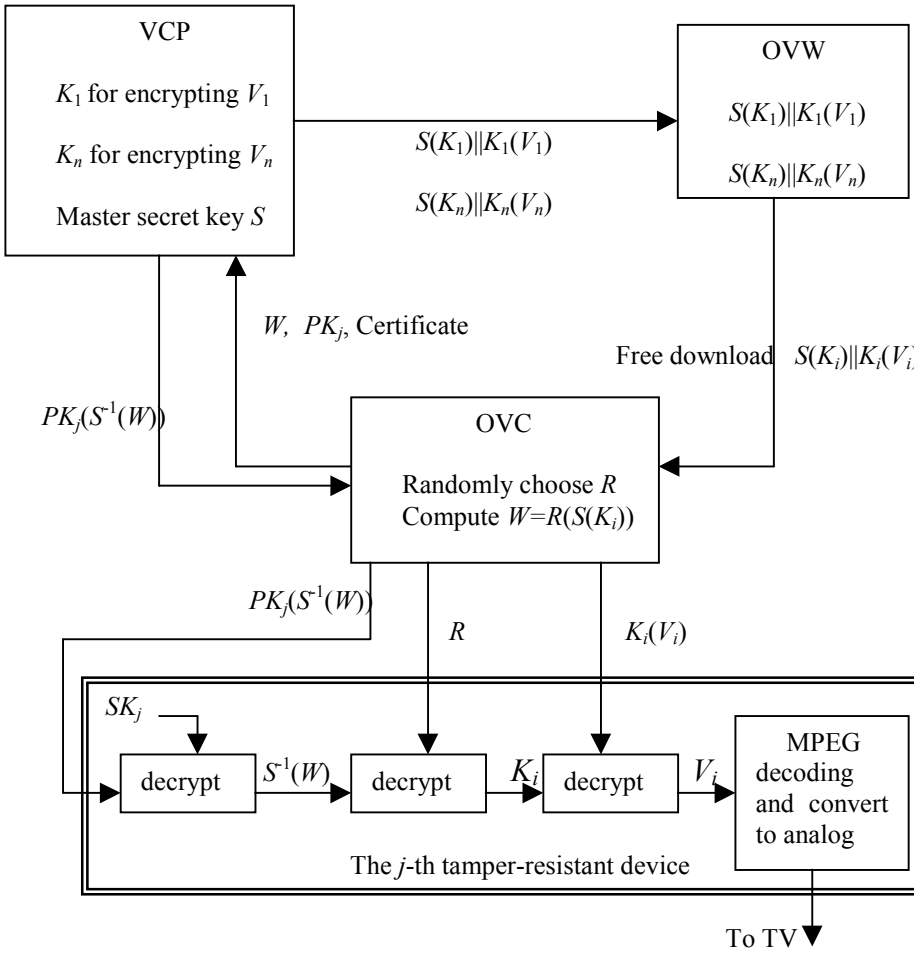
2.2 System Description in Detail

The system has three encryption algorithms:

1. Symmetric Key Cryptosystem I (SKC I)---SKC I is a fast cipher as studied in Section 6.
2. Symmetric Key Cryptosystem II (SKC II)---SKC II is a commutative cipher as studied in Section 4 and Appendix A.
3. Public Key Cryptosystem (PKC)---PKC can be any public key cryptosystem. A PKC with fast decryption as presented in Section 7 may be favored for this application.

System Description:

1. A VCP has n videos $V_1, V_2, \ldots, V_n$. The VCP chooses n secret keys $K_1, K_2, \ldots, K_n$ and encrypts $V_1, V_2, \ldots, V_n$ with SKC I, respectively. Denote the n ciphertexts by $K_1(V_1), K_2(V_2), \ldots, K_n(V_n)$.
2. The VCP also chooses a key S and encrypts $K_1, K_2, \ldots, K_n$ by S with SKC II. Denote the ciphertexts by $S(K_1), S(K_2), \ldots, S(K_n)$.
3. Suppose an OVC wants to watch V_i. He downloads $S(K_i)\| K_i(V_i)$ and chooses a key R for SKC II and encrypts $S(K_i)$. Denote the ciphertext by $W=R(S(K_i))$.
4. Decryption of W by key S is denoted by $S^{-1}(W)$.
5. PK_j/SK_j is a pair of public/private key generated by THM. SK_j is embedded into the j-th tamper-resistant hardware. PK_j is certified by THM and is given together with the certificate to the OVC who buys the hardware device.
6. When a VCP receives a PK_j, the VCP should check whether the PK_j is legal.



3 Content Protection

3.1 A Brief Review of Content Protection Technologies

Content protection is the key security issue for e-commerce of digital goods, no matter the transacted digital object is a picture, a video, an audio or a piece of news. It is commonly recognized that a digital content provider is hard to survive without certain means of protection. In online distribution of video, the content protection is the issue about how to prevent the illegal users (who do not pay) from watching a video. Content protection for digital goods is a very difficult problem from the technology angle. So far no fully satisfactory solution exists. Available technologies for content protection include follows.

Watermarking Technology

Watermarking technology has been considered to be a key technology for multimedia content protection. There have been so many research papers addressing watermarking technology in the past several years. Readers are referred to [CL97, CMY96, ZK95] and the references therein.

There are two sorts of watermarking. The first one is for ownership. The second one is for tracing illegal users. The technique of the second sort is also called fingerprinting in some references. The first sort of watermarking is to embed an identical watermark into every copy of the digital object. Hence, it cannot be used to distinguish who is the user who has distributed the illegal copy. The technology is to deter the large-scale resale. There are a lot of research publications in this area.

The second sort is to embed different watermarks into different copies. It can be used to trace the illegal users. But this sort of watermarking has certain difficulties. One is how to efficiently resist colluding attacks [BS95]. Another one is, as pointed out in [PS96], that there is actually no lawful basis for the content provider to sue the illegal user. This is because the provider himself possesses the watermarked digital object. Hence there is no way to distinguish who actually disclosed the copy. Asymmetric fingerprinting was proposed to solve this problem, see [PS99] and the references therein. However, it seems that the technique is not ready for practical use due to its complicated and interactive implementation.

Tamper-Resistant Software

This technology is to prevent the decryption party from accessing the decryption key in software. Combining with other techniques, the technology can be used to prevent making illegal copies. This is advantageous over watermarking technology at the point that watermarking is used to *catch* illegal copy while the tamper-resistant technology is used to *prevent* illegal copy. Tamper-resistant software, in principle, is to hide some secret information in a software program. It is based on anti-reverse-engineering. The current status of the technology is more like know-how and the technology is more studied within industry community than within academic community. There are quite a number of patents but rare publications in this area. It seems that there is no solid theoretical foundation for this technology.

Tamper-Resistant Hardware

Tamper-resistant hardware has been studied for many years. This technology has already been used in many realistic applications such as cable TV and DVD etc. In this paper, we take tamper-resistant hardware as our basis for content protection. The

tamper-resistant hardware in our system contains a private key that is used to decrypt the ciphertext of the secret key of a video.

There have been various attacks against tamper-resistant hardware devices, such as fault-differential attack [BDL97, JQBD97], timing attack, differential power attack [Kocher], and probing attack [AK97, HPS99], etc. Researchers find that any information leakage in the procedure of the computation may lead to the secret key compromised. On the other hand, various counter measures have been proposed. Counter measures against fault-differential attack can be found in [BDL97]. Methods to resist differential power attack are presented in [Cor99, Kocher]. [WBYD00] presents some counter measures against probing attack. But in general, the attitude toward tamper-resistant hardware from academic is negative. This is because the fact that it might be hard to absolutely prevent the leakage of side-channel information [CKN01], which would cause key-compromising.

Industry, whereas, has the different view on tamper-resistant devices, which have been running well in the reality. One example is cable TV box, which is insecure from whatever angle in researchers eyes. But it does make good business. There is a big gap between academic and industry in the recognition of security. The former tends to consider absolute security based on complexity assumptions while the latter usually concerns more about relative security with respect to the costs. We do not believe that tamper-resistant devices could be relied upon as the security basis for military or government secrecy. But we think it should be qualified for small-valued business. In addition, it is commonly recognized that tamper-resistant hardware is much more reliable than tamper-resistant software.

3.2 The Content Protection Based on Public Key Cryptosystem

Why Use Public Key Cryptosystem The most important and essential discipline for a content protection system is that component-compromising must not cause the whole system crashing. If we only use symmetric key cryptosystem in the tamper-resistant hardware devices, we have two choices. First, we can install a master secret key into every tamper-resistant hardware device. This choice is apparently not secure since breaking one device may cause the master key compromised, and therefore, the whole system is broken. The second choice is to install different secret keys into different devices. In this case all the VCPs must know all the secret keys (otherwise they cannot do encryption). This is also dangerous since once a VCP is compromised, all other content providers are exposed beyond any protection. The whole system crashes. Using public key cryptosystem perfectly solves above problems. The system proposed in this paper meets the discipline that component-compromising does not cause the whole system crashing.

Protection of Private Keys The private key installed in each tamper-resistant hardware device is very important. The manufacturer of the hardware (THM) must be very cautious on these private keys. A suggestion is to destroy the keys once they are installed into the hardware devices. The manufacturer is a trusted party like the CA in PKI. Actually, the manufacturer is required to maintain a revocation list as done by a CA. Once a device is found to be broken, its serial number should be put into the list to prevent its use any more.

Tamper-Resistant Technology In this paper we do not discuss how to build up tamper-resistant hardware devices. There has been research on this technique for

many years. What we want to emphasize here is that in our system the tamper-resistant technique can be focused on the private key. It is the critical clue. Once the private key is destroyed, the device is completely useless. So the guideline to build tamper-resistant property should be that *once the device is tampered or opened, the private key is automatically erased or modified***.** Protection techniques may include, for example, hiding a photoelectric cell inside the device, which is touched off (once the device is opened) to erase/change the private key. Another technique is a kind of careful wiring from inside so that the device is hard to be opened without breaking off the wire, which would also cause the private key erased. Of course, there must be multiple levels of protection.

Business Consideration In the proposed system, every OVC must buy a hardware device. This is the disadvantage of hardware solutions compared with software solutions. But from another angle, hardware solutions are not excluded here since a video has a comparatively high value. A DVD movie usually costs about 20 US dollars, while online video may cost much less as long as the content is perfectly protected. If an online video costs only one dollar for example, the attraction for a customer to buy a hardware device is considerably large. More specifically, such a tamper-resistant hardware device is not expensive since the processors to conduct decryption and D to A converting are not very expensive. Another choice is to build such hardware device into home appliances like VCD/DVD players. Then there is only a small additional cost while the player has a new function used together with the home PC. That is very alluring.

4 Privacy Protection

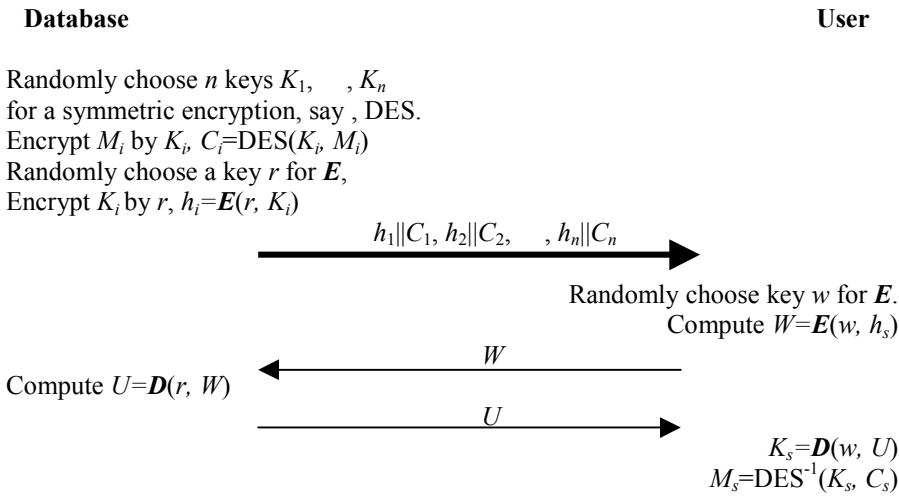
A VCP may provide a large number of videos with various categories. This is also the attractive point of online video. Therefore, privacy is another concerned issue. A customer may not like to let others know what video he/she is watching or is in favor. Such privacy should be guaranteed as long as the online video is a charged service. From another viewpoint, if two VCPs are providing same video at same price, the one who guarantees privacy is more competitive.To retrieve a message from a database without revealing which message is actually being retrieved has been theoretically studied under the term PIR (private information retrieval) [CGKS95, CG97, KO97, CMS99]. However, the computational costs of these solutions are very large due to their bit-by-bit processing manner. All those schemes need at least computation of $O(N)$ operations for retrieving only one bit, where N is the number of bits of the whole database. All the previous PIR schemes are aiming at reducing communication complexity. The scheme of [CMS99] can even achieve a communication complexity of poly(logN). However, those schemes can hardly be accepted for practical use. In this section we propose a simple and efficient scheme for privacy protection in the online video system. The scheme is not a PIR scheme from a strict viewpoint. We describe the scheme in the way as describing a PIR scheme for simplicity.

A Simple PIR Scheme

In Appendix A, we will present our scheme in detail and give cryptanalysis. Here only its principle is given. Let $\boldsymbol{E}$ be a symmetric-key encryption algorithm that has *commutative property*, i.e., for any pair of keys K_1, K_2 and for any message m, we have

$$\boldsymbol{E}(K_1, \boldsymbol{E}(K_2, m))=\boldsymbol{E}(K_2, \boldsymbol{E}(K_1, m))$$

Denote the decryption algorithm of $\boldsymbol{E}$ by $\boldsymbol{D}$. Suppose that the Database has n files denoted by M_1, M_2, , M_n (possibly with different lengths) and the User wants M_s. By the following scheme the User can get M_s without Database knowing what s is.



The Database has no way to know which message the User can get, no matter how maliciously Database performs. Meanwhile the User can only get one message by implementing the scheme once. In [BDF00], a concrete $\boldsymbol{E}$ and the security analysis of the protocol were presented.

It is easy to see that the above PIR scheme processes messages file-by-file. It does not get communication complexity reduced if it is regarded as a PIR scheme. But it fits our online video scheme very well because of the following reasons.

The customer s downloading can be anonymous. When the customer downloads the encrypted video from OVW, he need not show his personal information such as membership or credit card number etc. If the download is through some specific proxy, the customer's IP address can be hidden. Or if the download is through dial-up, the IP address changes every time. Some companies, such as www.zeroknowledge.com and www.anonymizer.com, provide service for anonymous download. On the other hand, the communication between the customer and the VCP cannot be anonymous since the VCP must know whom he is dealing with. When the VCP decrypts the secret key for the customer, the service is a charged service. Either the membership authentication or a payment is needed, which would disclose certain information about the customer.

5 System Features and Discussions

The system proposed in this paper has many good features.

1. The system is flexible. There may be multiple VCPs and OVWs. An OVW can support multiple VCPs. Each OVC can enjoy services from multiple VCPs with only one hardware device.
2. No VCP holds any secret of any OVC (the secret of his hardware device); therefore, if a VCP is compromised or becomes malicious, the other VCPs are not effected.
3. The OVCs' privacy is guaranteed no matter how malicious a VCP performs. At most a VCP can let an OVC receive no service, but can never get to know which video the OVC is trying to watch. On the other hand, VCP can still get statistical data on frequency of videos being downloaded from OVW (this seems necessary for business).
4. Low requirement on download speed. Unlike streaming VoD, where the network speed must be faster than the speed of video playing, in this system an OVC can download the encrypted video at the speed slower than that of video playing. The download can also be in off-peak hours.
5. Cheap computations. The system exploits some cryptosystems. But the crypto operations required in the system are light.

Compared with DVD The DVD encryption scheme is not robust: all the videos are encrypted by one secret key (for each zone) and the secret key is stored in all DVD players. Disclosing the secret key causes the whole scheme cracked. Our video distribution scheme is designed to be robust. In our scheme, the private keys in different hardware devices are independent from each other. In case one hardware player is cracked, the other hardware devices are not affected. Even if the hacker makes the cracked key public, the damage would be limited: the VCPs just refuse to provide service to that device any more.

Payment Choice and Privacy There are two ways of payment for online video. The first one is like membership. An OVC can subscribe to a VCP and the VCP will always serve the OVC (there may be a limit on number of videos for the OVC per month). For this payment manner, the OVC's privacy is perfectly protected. The second payment way is pay-per-video. For this payment manner, the privacy can only be guaranteed among all the videos with the same price. In this case the system needs a slight modification. The master key S in Section 2.2 should be replaced with a set of master keys, each key for one group of videos with same price.

Flexible Distribution Means In reality there may be more means to distribute the encrypted videos. The VCPs encourage the distribution of the encrypted videos among video fans. Another possibility is by CD. The CD with huge storage capability is going to emerge in a few years. We believe that the storage media is much cheaper than the stored content. A CD containing many encrypted movies can be very cheap in the future.

6 Fast Symmetric Key Encryption Scheme

It is well known that symmetric key encryption schemes are much faster than PKC schemes. For example, DES can achieve speed 20-30 Mb/s on a 233 MHz Pentium II Processor [Dai]. That speed is sufficient for video play. However, the decryption of the video may be conducted on a resource-limited chip. In our system, the speed of the processor in the tamper-resistant hardware device may be much slower than a 233 MHz Pentium II Processor. So it is better if we have faster encryption algorithms. We show that there is a large room to increase the speed of a symmetric key encryption algorithm while maintaining its security as long as the encrypted file is very large.

It is widely believed that there is a tradeoff between the speed and security strength of a cipher. It is a big challenge to design a very strong cipher that has very fast speed. But if we consider the situation of encrypting large files, it is possible to design a cipher with both very fast speed and very strong security. The reason is that we can combine a very strong but slow cipher with a very fast but weaker cipher such that the combined cipher is very fast and very strong. The reason behind the construction is that a weaker cipher may be a strong one if it is used in the way that each key is used to encrypt a limited amount of messages only. Just looking at those powerful cryptanalysis techniques, such as differential attack [BS91] and linear attack [Mat93], large amount of chosen/known plaintext/ciphertext pairs are always the pre-condition.

In our scheme we combine fast stream ciphers with secure block ciphers.

Let ***SE*** denote a strong encryption algorithm, such as AES, and ***FE*** be a weaker but very fast encryption algorithm, such as some fast stream-cipher. Denote a plaintext by $M_1M_2M_3 \quad M_n$ where M_i is a block of size same as that of ***SE***. Let K be a key, the encryption can be done as follows

$$\begin{aligned}\text{Ciphertext} = & \boldsymbol{SE}(K, M_1)\|\boldsymbol{FE}(K_1, M_2M_3 \quad M_k)\| \\ & \boldsymbol{SE}(K, M_{k+1})\|\boldsymbol{FE}(K_2, M_{k+2}\, M_{k+3} \quad M_{2k})\| \\ & \quad \| \\ & \boldsymbol{SE}(K, M_{tk+1})\|\boldsymbol{FE}(K_{t+1}, M_{tk+2}M_{tk+3} \quad M_n)\end{aligned}$$

where $K_{i+1}=\boldsymbol{SE}(K, \boldsymbol{SE}(K, M_{ik+1}))$ (i=0,1, ,t) are called segment keys. The k (segment size) is the value determined by ***FE*** such that ***FE*** is strong enough if one key is used to encrypt at most k blocks forever.

It is obvious that such combinations have speed advantage only for large files. If the plaintext consists of only a few blocks, the speed is close to that of ***SE***. But if the plaintext is large and the k is fairly large, the speed is close to that of ***FE***. In analogy this is like to construct a door with steel frame and plastic filling pieces such that the door is as light as a plastic door while as strong as a steel door. The k is like the size of the grid. The smaller it is, the more secure the scheme.

Dividing the video into segments is also needed for fast-preview. The video can be played from any segment. In Appendix, we show two concrete ciphers with speed 300 Mb/s, and 1,500 Mb/s on a 233 MHz Pentium II Processor.

We have seen some research papers, such as [MS95, QNT97, Tan96 etc] on increasing video encryption speed by exploiting the structures of MPEG. But none of them can compare with our solution. Ours is very much faster as long as the encrypted file is large.

7 PKC with Fast Decryption

In our system, a tamper-resistant hardware device contains a private key of a PKC (public key cryptosystem). The PKC decryption is conducted in the device. Although any PKC can be used in our system, a PKC with fast decryption is favored for lowering the cost of the device. It is well known that RSA can be made fast for encryption. But PKC with fast decryption has rarely been studied. In this section we make an effort to design a PKC with fast decryption. We propose a PKC that is much faster in decryption than RSA and at least ten times faster than MultiPrime, while the security strength is the same. The PKC proposed here is similar to Shamir s unbalanced RSA except that we have a small d. In RSA a small d is dangerous. We show that our scheme is immune to small d attack. To our knowledge, this is the first PKC design for fast decryption.

Algorithm Description

Private key: primes p, q (better p, q are safe primes) and an odd number d .
Public key: $n(n=pq)$, $e(ed\equiv 1 \bmod q-1)$.
Encryption: $c=m^e \bmod n$ where m $(0<m<q)$ is the plaintext, c is the ciphertext.
Decryption: $m=c^d \bmod q$

It is easy to verify that the decryption is correct. The scheme is different from RSA at the point there is an expansion from plaintext to ciphertext.

Fast Decryption

We take $|n|$=1024, $|q|$=341 and $|d|$=120. The decryption speed of this scheme is apparently much higher than 1024 bit RSA. But the most important issue is the security. It is dangerous to take small d in RSA. In our algorithm, however, a small d is conjectured to be safe.

Security Analysis

Small *d*. It is shown in [Wie90] that if d is small, say $|d|<|n|/4$, then the RSA scheme can be broken. The attack is very simple and beautiful:

> In number theory we have: if η/ξ is an approximation of a known number c within $1/\xi^2$, i.e., $|c-\eta/\xi|<1/(2\xi^2)$, then η and ξ can be efficiently computed out by continuous fraction. Since $ed\equiv 1 \bmod \varphi(n)$, we have $ed=k\varphi(n)+1$ for some k, $|k|\le|d|$. Then $|e/n-k/d|=|(kp+kq-k+1)/(nd)|<1/(2d^2)$ due to $|d|<|n|/4$. Therefore, k and d can be quickly computed from e and n.

In [BD99], the result is improved to breaking RSA for $|d|<0.292|n|$ by lattice method, which can be regarded as the generalization of approximation in multiple dimensions. In both [Wie90] and [BD99], the e satisfying $ed\equiv 1 \bmod \varphi(n)$ is the key point. But in our scheme, the e satifies $ed\equiv 1 \bmod q-1$ instead of mod $\varphi(n)$. If we target at the d' such that $ed'\equiv 1 \bmod \varphi(n)$ for public key e, the d' must be very large. Another attack to small exponent d is the meet-in-the-middle attack that is similar to the birthday attack but requires FFT technique. The complexity of that attack is $O((\log d)^2\sqrt{d})$. Therefore taking d 120 bits gives a security level of $2^{70\sim 80}$.

Chosen Ciphertext Attack. The scheme is fragile to chosen ciphertext attack. An attacker can choose a $M>q$ and set $c=M^e \bmod n$. The decryption $m=c^d \bmod q$ satisfies $\gcd(n, M-m)=q$. However this attack does not cause any problem if we carefully

choose a mapping format before encryption, as done in [BR94] and [OU98], which provide provable security. Besides, the application of the scheme in our video system prevents the chosen ciphertext attack since the decrypted value never goes out of the tamper-resistant hardware device. The decrypted value is the key to encrypt video. So chosen ciphertext attack does not apply.

Factorization. In our scheme, n is a composite of two primes with different sizes. For the situation where n has 1024 bits and the smaller prime factor has 341 bit, the current factoring techniques cannot provide better performance than factoring 1024-bit n with two equal-size primes. This is because the most efficient number field sieve algorithm has complexity $L_n(1/3, c)$, which is dependent on size of n. Elliptic curve factoring algorithm is dependent on the size of the smaller prime q, but it has complexity $L_q(1/2, c)$. So currently available factoring techniques do not make factoring our n easier. The same argument is taken in [OU98], where $n=p^2q$ has 1024 bits, and in MultiPrime [Compaq] where n ($|n|$=1024) is a composite of three different primes.

References

[AK97] R. Anderson and M. Kuhn, "Low cost attacks on tamper resistant devices", in Security protocols: International Workshop'97, LNCS 1361, Springer-Verlag, pp.125-136, 1997.

[BDF00] F. Bao, R. Deng, P. Feng, An efficient and practical scheme for privacy protection in e-commerce of digital goods , Pre-Proceedings of The 3rd International Conference on Information Security and Cryptology (ICISC00), pp. 167-176, 2000.

[BR94] M. Bellare and P. Rogaway, "Optimal asymmetric encryption", Eurocrypt'94, LNCS, Springer-Verlag, 1995.

[BS91] E. Biham and A. Shamir, "Differential cryptanalysi of DES-like cryptosystems", Journal of Cryptology, Vol. 4, No. 1, pp. 3-72, 1991.

[BAK] E. Biham, R. Anderson and L. Knudsen, "Serpent: a proposal for the advanced encryption standard", http://www.cl.cam.ac.uk/~rja14/serpent.html.

[BD99] D. Boneh and G. Durfee, "Cryptanalysis of RSA with private key d less than $N^{0.292}$", Eurocrypt'99, pp. 1-11, Springer-Verlag, 1999.

[BDL97] D. Boneh, R. DeMillo, and R. Lipton, "On the importantance of checking cryptographic protocols for faults", in Proc of Eurocrypt'97, LNCS 1233, Springer-Verlag, pp. 37-51, 1997.

[BS95] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data", Crypto'95, LNCS, pp. 452-465, Springer-Verlag, 1995.

[Cor99] J. Coron, Risistance against differential power analysis for elliptic curve cryptosystems , Proc. Of CHES 99, LNCS 1717, Springer-Verlag, pp. 292-302, 1999.

[CG97] B. Chor and N. Gilboa, "Computational private information retrieval", Proc. of 29th STOC, pp. 304-313, 1997.

[CGKS95] B. Chor, O. Goldreich, E. Kushilevita, and M. Sudan, "Private information retrieval", Proc. of 36th FOCS, pp. 41-50, 1995.

[CKN01] J. Coron, P. Kocher, and D. Naccache, Statistics and secret leaksge , to appear in the Proceedings of Financial Cryptography 01, LNCS, Springer-Verlag.

[CL97] I.J.Cox, J.P.M.G.Linnartz, "Some general methods for tampering with watermarks", in IEEE international Conference on Image Processing", 1997.

[CMS99] C. Cachin, S. Micali, and M. Stadler, "Computationally Private Information Retrieval with Polylogrithmic Communication", in Proceedings of Eurocrypt'99, LNCS, Springer-Verlag, pp. 402-414, 1999.

[CMY96] S.Craver, N. Memon, B. Yeo, M. Yeung, "Can invisible watermarks resolve rightful ownership", IBM Research Report, RC 20509, July 25, 1996.

[CNS99] J. Coron, D. Naccache and J. Stern, "On the security of RSA padding", Crypto'99, pp. 1-18, Springer-Verlag, 1999.

[Compaq] http://www.tandem.com/brfs_wps/esscpttb/esscpttb.htm

[CWSK98] D. Coppersmith, D. Wagner, B. Schneier and J. Kelsey, "Cryptanalysis of TWOPRIMES", Proceedings of FSE'98, LNCS, Springer-Verlag, 1998.

[Dai] W. Dai, "Speed benchmarks of various ciphers and hash functions", http://www.eskimo.com/~weidai/benchmarks.html.

[GIKM98] Y. Gertner, Y. Ishai, E. Kushilevita and T. Malkin, "Protecting data privacy in private information retrieval schemes", Proc. of 30th STOC, 1998.

[GMDS98] C. Griwodz, O. Merkel, J. Dittmann, R. Steinmetz, Protecting VoD the Easier Way , ACM Multimedia 98, pp. 21-28, Bristol, UK, 1998.

[HPS99] H. Handschun, P. Paillier, and J. Stern, Probing attacks on tamper-resisyant devices , Proc. Of CHES 99, LNCS 1717, Springer-Verlag, pp. 303-315, 1999.

[Jai99] R. Jain, "The convergence of PCs and TV", IEEE Multimedia, October/December 1999.

[JQBD97] M. Joye, J.-J. Quisquater, F. Bao, and R.H. Deng, "RSA-type signatures in the presence of transient faults", In M. Darnell, editor, Cryptography and Coding, Vol. 1355 of Lecture Notes in Computer Science, pp. 155--160, Springer-Verlag, 1997.

[Knu98] L. R. Knudsen, "The block cipher lounge---AES", http://www.ii.uib.no /~larsr/aes.html.

[KO97] E. Kushilevita and R. Ostrovsky, "Single-database computationally private information retrieval", Proc. Of 38th FOCS, 1997.

[Kocher] P. Kocher, http://www.cryptography.com/resources/

[Mat93] M. Matsui, "Linear cryptanalysis method for DES cipher", Proceedings of Eurocrypt'93, LNCS 765, Springer-Verlag, pp. 386-397, 1994.

[Minoli] Video Dialtone Technology, McGraw-Hill, 1995.

[MQ95] B. M. Macq and J-J Quisquater, Cryptology for digital TV broadcasting , Proceedings of the IEEE, Vol. 83, No, 6, pp. 944-957, 1995.

[MS95] T. Maples and G. Spanos, "Performance study of a selective Encryption scheme for security of networked, real-time video", Proc. of the 4th International Conference on Computer and Communications and Networks, Las Vegas, Nevada, Sept, 1995.

[OU98] T. Okamoto and S. Uchiyama, "A new public-key cryptosystem as secure as factoring", Proceedings of Eurocrypt 98, LNCS, Springer-Verlag, 1998.

[PS96] B. Pfitzmann and M. Shunter, "Asymmetric fingerprinting", Eurocrypt'96, LNCS 1070, pp. 84-95, Springer-Verlag, 1996.

[PS99] B. Pfitzmann and A. Sadeghi, "Coin-based anonymous fingerprinting", Eurocrypt'99, pp. 150-164, Springer-Verlag, 1999.

[QNT97] L. Qiao, K. Nahrstedt and I. Tam, "Is MPEG encryption using random lists instead of Zig Zag Order", IEEE International Symposium on Consumer Electronics, Dec, 1997.

[Rue86] R. A. Rueppel, Analysis and Design of Stream Ciphers, Springer-Verlag, 1986.

[Tan96] L. Tang, "Methods for Encrypting and decrypting MPEG video data efficiently", Proc. of the 4th ACM Multimedia Conference, Boston, MA, November, 1996.

[Wie90] M. Wiener, "Cryptanalysis of short RSA secret exponents", IEEE Transactions on Information Theory, Vol. 36, No. 3, pp. 553-558, 1990.

[WBYD00] H. Wu, F. Bao, D. Ye, R. Deng, "Cryptoanalysis of the m-permutation protection schemes", Proc. of ACISP2000, LNCS 1841, Springer-Verlag, pp. 97-111, 2000.

[ZK95] J. Zhao and E. Koch, "Embedding robust label into images for copyright protection", Proceedings of the International Conference on Intellectual Property Rights for Specialized Information, Knowledge and New Technologies, Austria, Aug. 21-25, 1995.

Appendix

Fast Encryption Scheme I

Now we introduce the first scheme of our fast encryption framework. AES is supposed to be the encryption standard for this whole century. It is regarded unbroken unless some impossible breakthroughs in math take place. Therefore, we take ***SE***() as AES (Rijndael). The stream cipher ***FE***() is given as follows. The whole picture of the scheme is described in Section 6.

Description of the Stream Cipher *FE*()

The stream cipher has a 128-bit key size and operates on 32-bit plaintext strings $b_1 b_2 \cdots b_m \cdots$. Denote the 128-bit key as $k = k_1 k_2 k_3 k_4$, where k_i s are 32-bit strings. Define

$$F(k,x) = ((((x + k_1) \oplus k_2) \times k_3) \oplus k_4) \hookleftarrow$$

where x is a 32-bit string, $\oplus$ is the bit-wise XOR, + and $\times$ are mod 2^{32} addition and multiplication respectively, and $\hookleftarrow$ is to reverse the 32 bits into opposite ranking. Encryption of the plaintext strings $b_1 b_2 \cdots b_m \cdots$ is then given by

$$d_i = b_i \oplus F(k, F(k, F(k, d_{i-1}) \oplus b_{i-1}) \oplus d_{i-2})$$

where $d_1 d_2 \cdots d_m \cdots$ are the corresponding ciphertext strings and where d_{-1}, d_0, b_0 are set to k_2, k_3, k_4, respectively.

We implemented the encryption scheme on a 233MHz Pentium-II/MMX processor (The encryption speed of Serpent on the same processor is about 25.8 Mbit/s). The experiment are given in the table below.

Table 1. Experiment Results of Encryption Scheme I

Total Data Size (bits)	Segment Size (bits)	Test 1 (Mbit/s)	Test 2 (Mbit/s)	Test 3 (Mbit/s)	Test 4 (Mbit/s)	Average Speed (Mbit/s)
5,242,880,000	32,768	297.0	296.1	297.0	297.0	296.7
5,242,880,000	65,536	304.0	302.9	304.0	304.0	303.7
5,242,880,000	131,072	307.0	307.0	307.0	308.1	307.3
5,242,880,000	262,144	309.1	309.1	309.1	309.3	309.2

Security Discussion

Security of the secret key: The secret key K is protected by ***SE***() which by our assumption, is secure against all known attacks.

Meet-in-the-middle attack to segment keys: This is a type of brute force attack. By meeting one or more bits in the middle, the attacker exhaustively search the key bits relevant to these middle bits. Since we take 3 rounds of F, the meet-in-the-middle attack does not work. This is because at least one of the two sides of the middle bits goes through two rounds of F; therefore, at least 96 bits of the key effect one middle bit.

Chosen ciphertext attack to segment keys: It is well known that all stream ciphers that have ciphertext feedback are vulnerable to chosen ciphertext attacks. Suppose our stream cipher was defined as

$$d_i = b_i \oplus F(k, F(k, F(k, d_{i-1}) \oplus d_{i-2}) \oplus d_{i-3}).$$

By letting $d_{i-1} = d'_{i-1}$, $d_{i-2} = d'_{i-2}$ and d_{i-3} and d_{i-3} differing in only one bit, an attacker can ask for the decryption of d_i and d_i by applying the **differential attack**. However, our stream cipher is defined by

$$d_i = b_i \oplus F(k, F(k, F(k, d_{i-1}) \oplus b_{i-1}) \oplus d_{i-2})$$

which has both ciphertext and plaintext feedback. In this case, if the attacker chooses both plaintext and ciphertext, the decrypted plaintext from the chosen ciphertext will have a very small chance to match the chosen plaintext. On the other hand, if the attacker tries to find such match from known plaintext/ciphertext (instead of chosen plaintext/ciphertext), the required number of known plaintext/ciphertext pairs is around 2^{48} blocks (like the birthday attack to $2^{3\times32} = 2^{96}$). However, this amount of plaintext/ciphertext pairs will not be available to the attacker since our segment size can never be so large.

Fast Encryption Scheme II

This encryption scheme is identical to Scheme I except that it uses another very fast stream cipher ***FE***(), which is given below.

Description of the Stream Cipher *FE*()

This stream cipher is used to expand a 128-bit key into a key stream of a plaintext segment size. Before illustrating its detailed design, we introduce the notations below:

- T: a table containing 19 elements, with each element consisting of 32 bits.
- T_i: the ith element of the table T
- k: the 128-bit secret key, consisting of four 32-bit words: k_0, k_1, k_2 and k_3.
- c_i: a 32-bit constant generated from the constant e as $c_i = (e \times 2^{32(i+1)})$ & 0xFFFFFFFF, i = 0 to 18.
- r'_i: a constant between 3 and 14. It is generated from the constant π as $r'_i = ((\pi \times 2^{8(i+1)}) \,\&\, 0\text{xFF}) \bmod 12 + 3$, i = 0 to 18.

We use the standard notations &, $\oplus$ and >>> to represent bit-wise AND, bit-wise XOR and right rotation, respectively. In addition, we define a feedback function F and an output function G below.

Definition of *F*. The input to F is the table T and a rotation constant r. The output of F is given as

$$f = ((T_0 \oplus T_4) + T_{13}) >>> r) \oplus T_{14}$$

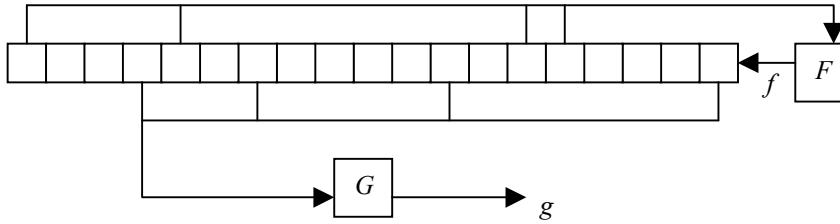
Definition of *G*. The input to G is the table T and the output is given as

$$g = ((T_{18} + T_{11}) \oplus T_6) + T_3 .$$

The operation of this stream cipher consists of two stages: the initial setup stage and the output stage or the main algorithm.

The initial setup stage

1. Let $T_i = c_i + k_{i \bmod 4}$, for $i = 0$ to 18.
2. Let $r_i = r_i' + ((k_0 >> 4i) \,\&\, 0xF)$ for $i = 0$ to 7;
   $r_i = r_i' + ((k_1 >> (4\times(i-8))) \,\&\, 0xF)$ for $i = 8$ to 15;
   $r_i = r_i' + ((k_2 >> (4\times(i-16))) \,\&\, 0xF)$ for $i = 16$ to 18.
3. Run the main algorithm below for 38 cycles and prepare for the output.



The main algorithm: For the ith cycle

1. Run the *F* function with $r = r_{i \bmod 19}$ to obtain the value of *f*.
2. Let $T_j = T_{j+1}$, $j = 0$ to 17, and let $T_{18} = f$.
3. Run the function G and generate the output g.

We implemented the encryption scheme described on a 233MHz Pentium-II/MMX processor. The encryption speed of Serpent on the same processor is about 25.8 Mbit/s. The experiment results are given in the table below.

Table 2. Experiment Results of Encryption Scheme II

Total Data Size (bits)	Segment Size (bits)	Test 1 (Mbit/s)	Test 2 (Mbit/s)	Test 3 (Mbit/s)	Test 4 (Mbit/s)	Average Speed (Mbit/s)
24,903,680,000	38,912	1234.9	1236.2	1234.9	1235.5	1235.4
24,903,680,000	79,824	1370.5	1370.5	1369.7	1370.5	1370.3
24,903,680,000	159,648	1448.8	1447.8	1451.8	1453.8	1449.8
24,903,680,000	319,296	1493.1	1492.1	1491.2	1493.1	1492.4

Security Discussion of Encryption Scheme II

First of all, the secret encryption key *K* is protected by ***SE***. Therefore, attacking the key is as hard as attacking AES, which is supposed to be absolutely secure against all attacks. Second, each segment key generated by ***SE*** is used to encrypt a plaintext

segment of very limited length by the stream cipher. For known ciphertext attack, our stream cipher can resist a large number of known ciphertexts.

The security of this stream cipher greatly depends on the feature that those 19 elements of the table T are updated in a non-linear way as the encryption goes on. With the carefully chosen parameters of function F, we can show that any two outputs of F are generated from at most one of the same elements of T. We note that each updated element, which is the output of F, is the non-linear combination of four elements of T. The key-related rotation amount in F strengthened the cipher further. With these unknown rotation amounts, we believe that it would be very difficult to find linear relationship among the elements of T.

The key stream is also generated from the elements of T in a non-linear way. The parameters of function G are carefully chosen so that any two outputs of G are generated from at most one of the same elements of T, and any output of G is generated from at most one of the same elements of T as any output of F (the updated element). Thus recovering the continuously updating elements of T from the output of G or revealing the linear relationship among the generated key stream becomes almost infeasible.

In this stream cipher, the elements of T are modified in a non-linear way. Thus it is not possible to compute the period of the generated key stream cipher. However, the period would not be a problem here. There are 19 32-bit elements of T. It is very unlikely that those elements will come back to their initial values even in the process of generating a 2^{128}-bit key stream. Furthermore, the stream cipher is used to encrypt only one package. The period of the output key stream is believed to be far larger than the size of a package.