

사용 시간에 따른 소프트웨어 요금 계산 기법

주한규^o 김동욱

한림대학교 컴퓨터공학과

hkjoo@hallym.ac.kr kdo309@hallym.ac.kr

Charge Calculation Scheme for Pay-Per-Use Software

요 약

대부분의 소프트웨어 사용자는 상용의 소프트웨어를 사용하기 위해서는 그 소프트웨어를 구입하여야 한다. 하지만 해당 소프트웨어를 많이 사용하지 않는 경우, 그 소프트웨어 비용은 과도한 지출로 인식된다. 이 경우 소프트웨어 대여가 효과적인 대안일 수 있다.

소프트웨어 대여를 지원하기 위해서는 요금 부과 기법이 필요하다. 두 종류의 요금 부과 기법이 생각될 수 있다. 하나는 정해진 기간에 일정 요금을 부과하는 것이며 다른 하나는 실제 사용 시간에 따라 요금을 부과하는 것이다.

이 논문에서는 소프트웨어의 실제 사용 시간에 따라 요금을 부과하는 기법을 제안한다. 소프트웨어의 실제 사용 시간에 따라 요금을 부과하기 위하여 사용 시간을 정확하게 측정할 수 있는 기법을 고안하였다.

1. 서론

대부분의 소프트웨어 사용자는 상용의 소프트웨어를 사용하기 위해서는 그 소프트웨어를 구입하여 사용한다. 많은 소프트웨어는 상당히 비싸다. 특히 사용자가 구입한 소프트웨어를 많이 사용하지 않는 경우, 사용자는 그 소프트웨어가 과도하게 비싸다고 인식하게 된다. 또한 이러한 경우 소프트웨어의 불법 복제 및 사용이 발생하게 된다.

소프트웨어 구입에 대한 문제점의 대안으로 소프트웨어 대여(임대)를 생각할 수 있다. 소프트웨어 사용자는 소프트웨어를 구입하는 대신 소프트웨어 제공자로부터 소프트웨어를 대여하여 사용할 수 있을 것이다. 소프트웨어 제공자는 소프트웨어를 대여하여 줄 수 있는 자격을 가진 사업자로 소프트웨어 개발자로부터 소프트웨어를 대여하여 줄 수 있는 권리를 획득한 사업자이다.

고속 네트워크의 발전으로 소프트웨어 제공자로부터 소프트웨어를 다운로드 받는 것은 많은 비용을 필요로 하지 않는다. 소프트웨어 사용자는 소프트웨어 제공자로부터 필요한 소프트웨어를 다운로드 받아서 사용하여 소프트웨어 제공자는 사용자에게 요금을 부과한다.

소프트웨어 대여 체계를 지원하기 위해서는 요금 부과 기법이 필요하다. 두 종류의 요금 부과 기법이 생각될 수 있다. 하나는 정해진 기간에 일정 요금을 부과하는 것이며 다른 하나는 실제 사용 시간에 따라 요금을 부과하는 것이다. 정해진 기간에 일정 요금을 부과하는 방식은 이미 상용으로 서비스되고 있다[1]. 이 기법은 구현하기 간단하나 해당 소프트웨어를

조금만 사용하는 사용자의 경우도 동일한 요금을 지불해야 하는 문제가 있다. 실제 사용 시간에 따라 요금을 부과하는 기법은 복잡한 방법이나 사용 시간에 비례하여 요금을 부과하므로 모든 사용자가 요금 부과에 납득할 수 있을 것이다.

실제 사용 시간에 맞추어 요금을 부과하기 위해서는, 사용 시간을 정확하게 측정할 수 있는 방법이 필요하다. 이 방법에 의하여 측정된 시간은 소프트웨어 제공자와 사용자가 모두 납득할 수 있어야 한다. 의도적이건 실수로 인해서이건 일방에 의하여 방해 받지 않고 정확하게 계산하며 또한 그 근거를 보일 수 있어야 한다. 실제 요금 부과는 소프트웨어 제공자에 의하여 이루어지므로 사용 시간 계산의 책임 또한 제공자에게 있다.

2. 관련 연구

BizmekaKanOffice[1]는 소프트웨어 임대를 하는 상용의 서비스이다. BizmekaKanOffice에서의 요금 부과는 일정 기간에 따라 정해진 요금을 부과하는 방식이다.

Ferreria 등[2]은 소프트웨어 사용에 따른 요금 부과 방식을 제안하였다. 그러나 [2]에서 제안된 기법은 소프트웨어의 수행 횟수에 따라 요금을 계산한다. 따라서 한 번 수행을 시작하고 오랜 시간 동안 종료하지 않고 사용하는 경우 짧은 시간만 사용하는 경우와 동일한 요금이 부과된다.

3. 서비스 모델

사용 시간에 따른 요금 부과 기법을 사용하는 소프트웨어 임대 방법은 3단계로 이루어져 있다. 사용자 등록,

소프트웨어 사용, 그리고 요금 부과 및 이의 해소가 그들이다.

새로운 고객이 소프트웨어 제공자가 제공하는 소프트웨어를 임대하여 사용하고자 하는 경우, 고객은 먼저 사용자 등록을 하여야 한다. 고객은 자신의 신상 정보, 요금 청구를 위한 주소, 공개키 인증서 등의 정보를 제공자에게 전송하여 사용자 등록을 한다. 제공자는 사용자를 위한 계정을 새로 생성한다.

등록된 고객이 소프트웨어를 사용하기 위해서는 제공자의 온라인 카탈로그를 참조하여 사용 가능한 소프트웨어와 그 임대 가격을 확인한다. 필요한 소프트웨어 사용을 결정하면 고객은 소프트웨어 제공자에게 사용 요청을 한다. 사용 요청을 받은 제공자는 소프트웨어를 제공한다 - 다운로드를 허용한다. 사용자는 소프트웨어를 필요한 만큼 사용하고 종료한다. 사용자는 종료 시 제공자에게 종료 메시지를 보낸다.

여러 이유로 종료 메시지가 제공자에게 전달되지 않을 수도 있다. 예를 들면 네트워크 문제로 종료 메시지가 손실될 수도 있다. 또한 사용자가 의도적으로 또는 실수로 종료 메시지를 보내지 못하는 경우도 있다. 소프트웨어 사용 도중 컴퓨터의 전원 문제로 사용이 중지될 수도 있다. 경우에 따라서는 사용자가 종료 메시지를 보내고 난 이후에도 계속해서 소프트웨어를 사용하고자 하는 사용자도 있을 수 있다.

소프트웨어 제공자는 사용자에게 사용 요금을 부과한다. 계산된 요금은 사용자의 소프트웨어 사용 시간에 기반한다. 사용자의 부과된 요금에 동의하면 요금을 지불한다. 만약 부과된 요금에 이의가 있는 경우 사용자와 제공자간에 논란이 생기게 된다. 논쟁이 생기는 경우 요금 계산이 제공자에 의하여 수행되었으므로 제공자가 그 요금이 올바른 요금임을 증명하여야 한다.

4. 제안 기법

4.1 접근 방법

입회자라고 불리는 프로그램과 반복 해쉬 함수가 사용된다. 입회자 프로그램은 간단한 프로그램으로 사용자의 컴퓨터에서 수행된다. 독립된 프로그램일 필요는 없으며 사용자가 대여하여 사용하고자 하는 소프트웨어에 내장되어질 수 있다. 입회자는 사용자의 부정행위를 탐지하고 부정행위 시 소프트웨어 사용을 중지시키는 역할을 한다. 입회자는 사용자의 컴퓨터에 어떠한 악영향을 끼치지 않으며 또한 어떠한 정보도 사용자의 컴퓨터로부터 외부로 보내지 않아야 한다.

입회자 프로그램은 사용자가 임대 소프트웨어를 시작하고자 하면 실행되며 사용자가 그 소프트웨어의 사용을 종료하면 동시에 종료된다. 사용자가 소프트웨어 제공자의 허락 없이 소프트웨어를 사용하면 강제로 그 소프트웨어를 종료 시킨다.

반복 해쉬[3, 4, 5]는 사용자의 사용 시간을 계산하기 위하여 사용된다. MD5[6] 또는 SHA[7]와 같은 해쉬 함수, $H(x)$, 는 일방향함수의 특성을 가지고 있다. 즉, 주어진 메시지 n 으로부터 $h = H(n)$ 을 만족하는 h 는 쉽게 구할 수 있으나, 주어진 h 로부터 $h = H(n)$ 을 만족하는 n 을 구하는 것은 현실적으로 불가능하다. 반복 해쉬란 이러한 해쉬 함수를 반복적으로 수행함을 말한다.

사용자가 소프트웨어 임대를 요청할 때, 무작위로 큰 정수 n 을 선택하고 그 n 에 m 번 해쉬를 수행한다.

$$H^i(n) = H(H^{i-1}(n)), \text{ where } i = 1, 2, 3, \dots, m, \\ \text{and } H^0(n) = n.$$

사용자는 $H^0(n), H^1(n), \dots, H^{m-1}(n)$ 은 비밀스럽게 보관하고 $H^m(n)$ 은 전자 서명을 첨부하여 소프트웨어 제공자에게 전송한다. 소프트웨어 제공자는 사용자에게 소프트웨어를 제공하여 사용할 수 있도록 한다. 이 때 입회자 프로그램도 동시에 사용자의 컴퓨터에서 수행되어야 한다.

소프트웨어를 사용하는 동안 사용자는 주기적으로 소프트웨어 제공자에게 $H^i(n)$ 을 전송함으로써 계속 사용하고자 하는 의사를 전달한다 - 여기에서 i 는 $m-1, m-2, \dots$ 로 매 번 1 씩 감소한다. $H^i(n)$ 을 수신한 소프트웨어 제공자는 이전에 받은 $H^{i+1}(n)$ 을 이용하여 $H^{i+1}(n) = H(H^i(n))$ 을 확인하여 그 해쉬 값의 올바름을 판별한다. 제공자는 올바른 해쉬 값을 받았음이 확인되면 사용자에게 계속사용 허락 메시지를 전송한다. 사용자는 그 허락 메시지를 입회자에게 전송하여 자신이 그 소프트웨어를 계속 사용할 수 있는 허락을 제공자로부터 획득하였음을 알린다. 주기적으로 입회자 프로그램은 사용자에게서 전달되어지는 계속 사용 허락 메시지(제공자로부터 생성 됨)를 확인하여 사용자가 계속 사용할 권한이 있음을 확인한다. 입회자 프로그램은 올바른 계속 사용 허락 메시지를 받지 못하면 사용자에게 사용 권한이 더 이상 없는 것으로 간주하여 그 소프트웨어의 사용을 중단시킨다.

계속 사용 허락 메시지 또한 반복 해쉬 함수를 이용한다. 소프트웨어 제공자는 소프트웨어 사용 요청을 받은 후 임의의 큰 정수 w 를 선택하고 m 보다 작지 않은 수 p 를 선택하여 이를 p 번 해쉬를 수행하여 $H^p(w)$ 를 사용자에게 전송하여 입회자에게 전달되도록 하며 $H^0(n), H^1(n), \dots, H^{p-1}(n)$ 은 비밀스럽게 보관한다. 이 때 $H^p(w)$ 가 소프트웨어 제공자로부터 새로 생성된 것임을 입회자가 확인할 수 있어야 한다. 이를 위하여 새로운 정수 하나를 입회자가 생성하여 제공자에게 전송하며 제공자는 이 수와 $H^p(w)$ 를 함께 전자서명을 첨부하여 전송함으로써 확인할 수 있도록 한다. 매 번 사용 허락 메시지로 $H^j(n)$ 을 전송한다 - 여기에서 j 는 $p-1, p-2, \dots$ 로 매 번 1 씩 감소한다.

사용자가 소프트웨어 사용을 마치면 소프트웨어 제공자는 사용 시간을 계산한다. 제공자가 마지막으로 받은 해쉬 값이 $H^{m-k}(n)$ 이면 그 사용자의 소프트웨어 사용 시간은 k 만큼이 된다. 사용 시간에 대한 사용자의 이의 제기가 있는 경우, 제공자는 처음 받은 해쉬 값($H^m(n)$) 과 마지막 받은 해쉬 값($H^{m-k}(n)$), 그리고 $H^m(n) = H^k(H^{m-k}(n))$ 을 보임으로써 계산된 시간이 올바른 사용 시간임을 보인다.

4.2 표기법

프로토콜 기술을 위하여 다음의 표기법을 사용한다.

- S : 제공자(서버)
- C : 사용자(클라이언트)
- O : 입회자
- Id_C : C 의 계정 이름
- $H(n)$: 메시지 n 의 일방향 해쉬 함수
- $H^k(n)$: 메시지 n 의 반복 해쉬. $H^k(n) = H(H^{k-1}(n))$ and $H^1(n) = n$
- P_S : S 의 공개키
- S_S : S 의 개인키
- P_C : C 의 공개키
- S_C : C 의 개인키
- $S_k(X)$: 메시지 X 대하여 키 k (S_S or S_C)로 수행한 전자서명
- N_A : A (C, S , 또는 O)에 의하여 생성된 nonce (랜덤 정수)

4.3 프로토콜

(1) 사용 요청

C 는 S 에게 사용자 등록을 한 것으로 가정한다. C 는 온라인 카탈로그를 이용하여 사용 가능한 소프트웨어와 그 소프트웨어의 시간당 임대 가격을 확인한 후, 해당 소프트웨어를 사용하기로 결정하면 제공자에게 사용 요청 메시지를 보낸다. C 는 요금 부과 시간 단위 L 을 결정한다. C 는 또한 현재 세션에서의 선택된 소프트웨어를 사용할 최대 시간(최대 시간 단위의 수) m 을 선택한다. C 는 랜덤 정수 $R1$ 을 선택하고 이 정수를 m 번 반복 해쉬를 수행한다. 즉, $H^1(R1), H^2(R1), \dots, H^m(R1)$ 을 계산한다. C 는 $R1, H^1(R1), H^2(R1), \dots, H^{m-1}(R1)$ 을 비밀스럽게 보관한다. 그리고 다음의 프로토콜이 수행된다.

1. $C \rightarrow S: Id_C$
2. $S \rightarrow C: N_S$
3. $C \rightarrow S: Id_C, Title, Pr, L, m, H^m(R1), N_C, S_{S_C}(Id_C, Title, Pr, L, m, H^m(R1), N_S)$
4. $S \rightarrow C: Ob, SW, S_{S_S}(Ob, SW, N_C)$

5. $O \rightarrow C: N_O$
6. $C \rightarrow S: N_O$
7. $S \rightarrow C: p, H^p(R2), S_{S_S}(p, H^p(R2), N_O)$
8. $C \rightarrow O: p, H^p(R2), S_{S_C}(p, H^p(R2), N_O)$

C 는 자신의 계정 이름 (Id_C)을 S 에게 전송하며 프로토콜을 시작한다. 메시지 1을 C 로부터 수신하면 S 는 새로 생성한 정수(N_S)를 생성하여 C 에게 전송한다. S 는 C 를 위한 세션을 시작한다. 이 세션은 C 가 소프트웨어 사용을 완료하면 종료된다.

C 는 Id_C 와 사용하고자 하는 소프트웨어 이름($Title$), 단위 시간 당 소프트웨어 사용 가격(P), 시간 단위(L), 그리고 최대 시간 단위 수(m), 해쉬 값($H^m(R1)$), 그리고 새로 생성한 정수(N_C)를 S 에게 전송한다. C 는 또한 $Id_C, Title, Pr, L, m, H^m(R1)$, 그리고 N_S 에 전자서명하여 그 서명을 S 에게 전송한다.

S 는 C 로부터 메시지 3을 이용하여 소프트웨어 사용 요청을 받은 후, 전자서명을 확인하여 올바른 요청인지 확인한다. 만약 올바른 요청으로 확인되면, S 는 입회자 프로그램(Ob)과 요청된 소프트웨어(SW)를 C 에게 전송한다.

C 는 메시지 4를 받으면 전자서명을 확인하여 올바른 메시지임을 확인하고 Ob 프로그램을 시작시킨다(수행되는 동안 사용자 컴퓨터의 메모리에 상주한다.). O 는 수행되는 동안 사용자 컴퓨터의 메모리에 상주한다.

O 는 새로운 정수(N_O)를 생성하여 이를 C 에게 전송한다. C 는 N_O 를 S 에게 전달한다.

S 는 N_O 를 수신한 후 m 보다 큰 랜덤 정수 p 를 선택한다. S 는 또한 랜덤 정수 $R2$ 를 선택하여 이를 p 번 반복 해쉬를 수행한다. 즉, $H^1(R2), H^2(R2), \dots, H^p(R2)$ 를 계산한다. S 는 $R2, H^1(R2), H^2(R2), \dots, H^{p-1}(R2)$ 을 비밀스럽게 보관하고, $p, H^p(R2)$, 그리고 $\{p, H^p(R2), N_O\}$ 에 대한 전자서명을 C 에게 전송한다. C 는 이 메시지를 그대로 O 에게 전달한다.

O 는 $p, H^p(R2), S_{S_S}(p, H^p(R2), N_O)$ 를 가지고 있는 메시지를 수신한 후 전자서명을 확인하여 이 메시지가 S 로부터 생성된 메시지임을 확인한다. O 는 자신이 생성하여 도전 메시지로 보낸 N_O 와 S 가 생성한 해쉬 값($H^p(R2)$)에 생성된 전자서명을 확인하여 수신한 해쉬 값($H^p(R2)$)이 S 에 의하여 현재의 세션을 위하여 새로 생성되었음을 확인한다.

올바른 메시지임을 확인하면 O 는 요청된 소프트웨어의 수행을 시작한다.

(2) 소프트웨어 사용

1. $C \rightarrow S: Id_C, i, H^i(R1)$
2. $S \rightarrow C: j, H^j(R2)$
3. $C \rightarrow O: j, H^j(R2)$

임대된 소프트웨어를 사용하는 동안 주기적으로 C 는 S 에게 사용 중인 소프트웨어를 계속 사용하고 싶다는 의사를 알린다. 소프트웨어 사용 요청 프로토콜 수행 시 C 가 생성해 놓았던 해쉬 값이 계속 사용 요청을 위하여 사용된다. 매 시간 단위(L)마다 C 는 $H(R1)$ 를 S 에게 전송한다. 여기에서 i 는 $m-1, m-2, \dots$ 의 값을 순차적으로 가지게 된다.

S 는 C 로부터 $H(R1)$ 를 수신한 후, $H(H(R1)) = H^{*1}(R1)$ 임을 확인한다. 만약 $H(H(R1)) = H^{*1}(R1)$ 이면, S 는 C 가 해당 소프트웨어를 L 시간만큼 더 사용하고자 하는 의사가 있음을 안다. S 는 C 로부터 올바른 계속 사용 요청을 받았음을 확인한 후, 계속 사용 허락 메시지를 전송한다. 계속 사용 허락 메시지를 위해서는 소프트웨어 사용 요청 프로토콜 수행 시 S 가 생성해 놓았던 해쉬 값($H(R2)$)이 사용된다. S 는 C 로부터 받은 계속 사용 요청 메시지 확인 후, 즉시 ($H(R2)$)를 전송한다. 여기에서 j 는 $p-1, p-2, \dots$ 의 값을 가진다.

C 는 수신한 $H(R2)$ 를 O 에게 전달한다. O 는 $H(H(R2)) = H^{*1}(R2)$ 임을 확인한다. 만약 $H(H(R2)) = H^{*1}(R2)$ 이면, O 는 C 가 시간 단위 L 만큼 현재 사용 중인 소프트웨어를 계속 사용할 권리가 있음을 알 수 있다.

만약 S 가 C 로부터 올바른 사용 요청을 받지 못하면, S 는 C 가 해당 소프트웨어를 계속 사용할 의사가 없는 것으로 간주하여 현재 세션을 종료하고 사용 시간을 계산한다. 이 경우 S 는 계속 사용 허락 메시지를 전송하지 않으며 O 는 계속 사용 허락 메시지를 수신할 수 없다. O 는 올바른 계속 사용 허락 메시지($H(H(R2)) = H^{*1}(R2)$)를 수신하지 않으면 현재 수행 중인 소프트웨어를 적절한 종료 예고 메시지와 함께 종료한다. 또한 자신의 수행(O)을 종료한다.

(3) 종료

1. $C \rightarrow S: SIG_Term, Id_C, Title, N_C, N_S, S_{SC}(SIG_Term, Id_C, Title, N_C, N_S)$
2. $C \rightarrow O: SIG_Term$

종료 메시지는 생략 가능하다. C 는 종료 메시지 없이 현재 사용 중인 소프트웨어를 종료할 수 있다. C 가 명시적으로 종료 메시지를 보내고 싶은 경우, 위에 기술된 종료 메시지를 S 와 O 에게 전송할 수 있다. 이 경우 S 는 현재 세션을 종료하고 사용 시간을 계산한다. O 또한 이 경우 현재 수행 중인 소프트웨어를 종료하고 자신의 수행(O)을 종료한다.

(4) 요금 계산

C 가 소프트웨어 수행을 마치면 S 는 소프트웨어

사용에 대한 요금을 계산한다. C 가 의도적으로 소프트웨어를 종료하였을 수도 있고 의도하지 않고 소프트웨어가 종료되었을 수도 있다. 어떠한 경우이건 S 는 세션을 종료하고 요금을 계산한다. 요금 계산은 C 가 소프트웨어를 사용한 시간 단위의 수와 시간 단위 당 요금에 의하여 결정된다.

C 가 소프트웨어 사용을 종료하였을 때, S 가 마지막으로 수신한 계속 사용 요청 메시지의 해쉬 값이 $H^{m-k}(R1)$ 이면 S 는 C 가 해당 소프트웨어를 k 단위 시간 이상 사용 하였음을 알 수 있다. 만약 C 가 사용 시간(요금)에 대한 이의를 제기하는 경우, S 는 $H^k(H^{m-k}(R1)) = H^m(R1)$ 임을 보임으로써 C 가 최소한 k 단위 시간 만큼 해당 소프트웨어를 사용 하였음을 증명할 수 있다.

5. 토론

본 논문에서 기술된 기법을 따르면 사용자의 소프트웨어 사용 시간을 정확하게 계산할 수 있다. 제공자가 어떠한 경우에도 사용자의 사용 시간을 과다하게 계산할 수 없다. 만약 소프트웨어 제공자가 $H^{m-k}(n)$ 을 가지고 있고 처음 사용자로부터 받은 해쉬 값이 $H^m(n)$ 이라면 사용자는 k 만큼의 시간 단위를 사용한 것이 된다. 해쉬 함수는 일방향 함수이므로 제공자가 $H^b(n)$ 로부터 $H^a(n)$ ($여기에서\ b < a$) 을 생성하는 것은 불가능하다. 따라서 제공자가 가지고 있는 $H^{m-k}(n)$ 은 소프트웨어 사용자가 k 만큼의 시간 단위를 사용한 후 계속 사용 의사를 밝힌 것이므로 최소한 k 만큼의 시간 단위 동안 소프트웨어를 사용한 것이다.

소프트웨어 사용자 또한 제공자에게 알리지 않고 소프트웨어를 사용할 수 없다. 매 주기마다 입회자 프로그램은 사용자가 소프트웨어를 계속 사용할 권한이 있는지 (사용자에게서 허락 메시지를 받았는지) 확인하여 제공자의 허락 메시지가 없으면 소프트웨어를 종료한다. 허락 메시지 또한 제공자만이 생성할 수 있으므로 사용자가 제공자임을 가장할 수 없다. 특히 제일 처음의 해쉬 값이 입회자가 생성한 임시 정수와 함께 전자 서명 되도록 하여 사용자가 이전에 사용했던 해쉬 값을 재사용하는 것이 불가능 하도록 하였다.

이 기법은 최대 1 시간 단위(L)만큼의 부정확함이 있을 수 있다. 소프트웨어 사용자가 $H^{m-k}(n)$ 을 전송한 후 즉시 종료하였으면 k 만큼의 시간 단위를 사용한 것이며, 계속 사용 허락 메시지를 받아 그 시간 단위의 남아있는 시간을 소진하였으면 $k+1$ 만큼의 시간 단위를 사용한 것이 되나 제공자는 판별할 수 있는 방법이 없다.

이 기법은 다른 소프트웨어와 마찬가지로 소프트웨어 크래킹에 안전할 수는 없다.

6. 결론

소프트웨어 임대 서비스는 완전히 새로운 개념은 아니다. 이미 상용의 소프트웨어 임대 서비스가 존재한다. 본 논문에서는 소프트웨어를 임대한 사용자가 사용한 시간만큼 요금을 지불할 수 있는 기법이 제안되었다. 이를 위하여 사용한 시간을 어떠한 환경에서든 정확하게 계산할 수 있는 기술이 필요하며 이를 위하여 반복 해쉬를 사용하였다. 본 논문에서 제안된 기법은 소프트웨어 제공자에 의하여 사용 시간이 계산되며 사용자가 부인할 수 없도록 시간 계산의 근거를 제시한다.

Acknowledgements

본 연구는 산업자원부의 지역혁신 인력양성사업(KOTEF)의 지원으로 이루어졌습니다.

참고문헌

- [1] BizmekaKanOffice, <http://kanoffice.bizmeka.com>, 2007.
- [2] L. C. Ferreria, R. Dahab, M. P. Aragao, and J. A. P. Magalhaes, "Two approaches for pay-per-use software construction," *Proc. of the 2nd IEEE International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems*, pp. 184-191, 2000.
- [3] L. Lamport, "Password authentication with insecure communication," *CACM*, vol. 24, no. 11, pp. 770-772, 1982.
- [4] T. Pedersen, "Electronic Payments for small amounts," *Proc. of the Cambridge Workshop on Security Protocols, LNCS 1189*, pp. 59-68, 1996.
- [5] J. Zhou and K. Lam, "A secure pay-per view scheme for web-based video service," *Proc. of the 2nd International Workshop on Practice and Theory in Public Key Cryptography, LNCS 1560*, pp. 315-326, 1999.
- [6] R. Rivest, "The MD5 message digest algorithm," *RFC 1321*, IETF, 1992.
- [7] NIST, "Secure hash standard," FIPS 180-2, 2002.