부정한 참가자의 신분 확인이 가능한 일방향 해쉬 함수에 기반한 온라인 비밀 분산 방식

오 수 현, 김 승 주, 원 동 호
성균관대학교, 전기전자 및 컴퓨터 공학부

One-way hash function based on-line secret sharing which identifies all cheaters

Soo-Hyun Oh, Seungjoo Kim, Dongho won
School of Electrical and Computer Engineering, Sungkyunkwan Univ.
E-mail : {shoh, sjkim, dhwon}@dosan.skku.ac.kr
URL : http://dosan.skku.ac.kr

요약

비밀분산은 비밀정보의 관리나 multiparty 프로토콜, 그룹 암호방식등의 분야에서 매우 중요한 부분이다. 따라서 본 논문에서는 일방향 해쉬함수에 기반한 효율적인 온라인 비밀분산 방식을 제안하고자 한다. 제안하는 방식은 하나의 share만으로 여러 개의 비밀을 분산할 수 있고, 액세스 구조가 변하는 경우에 notice board에 공개된 값들만 변경하면 각 참가자들은 기존의 share를 그대로 사용할 수 있다. 또한 참가자들의 부정이 있는 경우 그 수에 관계없이 부정한 참가자의 신분을 밝혀낼 수 있으며, 기존의 방식보다 계산상 효율적이라는 장점이 있다.

1. 서 론

비밀분산 방식이란 비밀 s를 n명의 참가자에게 분산하여 허가된 참가자들의 협조에 의해서만 s를 복원하는 것이 가능하도록 하는 방식이다. 이는 비밀정보의 관리뿐만 아니라 은 multiparty 프로토콜이나 그룹 암호방식등의 분야에서 매우 중요하다.

1979년 처음으로 Shamir[8]는 n명의 참가자 중 t(<n)명 이상이 협조하면 s를 복원할 수 있는 다항식 보간을 이용한 (t, n) threshold 방식을 제안하였다. 그후 1988년에 Tompa, Woll[10]등은 Shamir의 방식이 참가자들의 부정에 공격당하기 쉽다는 것을 지적하고 이를 막을 수 있는 방식을 제안하였다.

그 후, Cachin[3] 등은 각 참가자들이 비밀정보와 비슷한 크기의 하나의 share만을 이용하여 여러 개의 비밀을 분산할 수 있고 액세스 구조가 변하는 경우에도 기존에 분산된 share를 변경하지 않아도 되는 온라인 비밀분산 방식을 제안하였다. 이 방식은 모든 참가자들이 접근할 수 있도록 notice board와 같이 공개된 곳에 인증된 공개 정보를 공개함으로써 구현될 수 있다. 그러나 이 방식에서는 하나의 비밀이 복원된 후에도 다른 비밀들이 안전하기 위해서는 각 참가자의 share가 다른 참가자들에게 드러나지 않도록 하기 위해 추가적인 distributed computation[13]이 필요하다.

따라서 Pinch[7] 등은 비밀복원 과정에 Diffie-Hellman 문제[4]를 이용하여 각 참가자들의 share가 직접 드러나지 않게 하여 이러한 추가적인 계산이 필요 없는 개선된 방식을 제안하였다. 그러나 이 방식 또한 비밀복원 과정에서 참가자가 자신의 share를 바르게 제공하지 않는 경우에, 부정한 참가자의 신분을 밝혀낼 수 없고 그 부정한 참가자만이 정확한 비밀을 복원할 수 있다는 문제점이 있다.

이를 해결하기 위해 Ghodosi[5] 등은 비밀복원 과정 이전에, dealer에 의해 notice board에 공개된 정보를 이용하여 각 참가자들이 자신의 share를 바르게 제공하는지를 검사할 수 있는 과정을 추가하여 참가자들 중 과반수 이상이 정직할 경우에는 올바른 비밀을 복원할 수 있는 방식을 제안하였다. 그러나 이 방식은 과반수 이상의 참가자의 공모에 대해서는 안전하지 않다. 따라서 Yeun[11][12] 등은 Pinch의 비밀복원 과정에 디지털 서명을 적용하여 부정한 참가자의 수에 관계없이 항상 그의 신분을 밝혀낼 수 있는 온라인 비밀분산 방식을 제안하였다.

본 논문에서는 일방향 해쉬함수에 기반한 효율적인 온라인 비밀분산 방식을 제안하고자 한다. 제안하는 방식은 ① 하나의 share만으로 여러 개의 비밀을 분산할 수 있고, ② 액세스 구조가 변하는 경우에 notice board에 공개된 값들만 변경하면 각 참가자들은 기존의 share를 그대로 사용할 수 있다. ③ 또한 참가자들의 부정이 있는 경우 그 수에 관계없이 부정한 참가자의 신분을 밝혀낼 수 있으며, ④ 기존의 방식보다 계산상 효율적이라는 장점이 있다.

2. Yeun 등의 비밀분산 방식

Yeun등은 Pinch의 비밀분산 방식을 개선하여 부정을 저지른 참가자의 수에 관계없이 모든 부정한 참가자의 신분을 밝혀낼 수 있는 방식을 제안하였다. 이 방식은 이산대수 문제(Ⅰ)[11]나 RSA 암호방식(Ⅱ)[12]을 이용하여 비밀을 복원하는 과정에서 참가자들의 비밀이 드러나지 않게 하였고 전송 정보에 디지털 서명을 하여 비밀이 제대로 복원되지 않은 경우에 dealer에 의해 부정한 참가자의 신분을 확인할 수 있도록 하였다. 프로토콜은 다음과 같다.

[시스템 설정]

- P : 비밀분산에 참여하는 참가자 $P_i(1 \le i \le n)$의 집합
- D : dealer 단, $D \notin P$
- Γ : access structure ($\Gamma \in 2^{|P|}$)
- Γ^* : minimal authorized set
- K : 분산할 비밀
- f : 충돌 회피성 일방향 해쉬함수
- S_{P_i} : 참가자 P_i의 디지털 서명

[비밀분산 프로토콜 (I)]

단계 1) Dealer는 큰 소수 p와 ord(g) = q (단, $q|p-1$)인 g를 선택하고 각 참가자의 비밀값 $S_i < q$ $(1 \le i \le n)$들을 랜덤하게 선택한다.

단계 2) Dealer는 S_i값들을 비밀리에 각 참가자에게 전송한 후, 참가자의 ID와 함께 그 값들을 안전하게 저장한다.

단계 3) Dealer는 $X \in \Gamma^*$에 대해 다음과 같이 T_X값을 계산한다.

$$T_X = K - f\left(g_x^{\prod_{x \in X} S_x} \right) \pmod p$$

단계 4) Dealer는 각 Γ^*에 속하는 각 원소 X에 대해 (g_X, T_X)와 f(K) 값을 notice board에 공개한다.

[비밀복원 프로토콜(I)]

비밀을 복원하기 위한 필요한 참가자의 집합을 $X = \{P_1, P_2, ..., P_t\}$라 하자.

단계 1) 참가자 P_1은 X에 해당하는 g_X, T_X, f(K)를 notice board로부터 읽어온다.

단계 2) P_1은 $g_X^{S_1} \bmod p$를 계산하고 다음과 같이 디지털 서명을 생성한다.

$$S_{P_1} = Sig_{p_1}(g_X^{S_1} \bmod p \,\|\, X \,\|\, g_X)$$

단계 3) P_1은 S_{P_1}과 $g_X^{s_1} \bmod p$를 P_2에게 전송한다.

단계 4) 각 $P_i(1 < i < t)$는 P_{i-1}로부터 받은 서명을 검증한 후 $(g_X^{s_1 s_2 \cdots s_{i-1}})^{s_i} \bmod p$를 계산하고 다음과 같이 디지털 서명을 생성하여 P_{i+1}에게 S_{Pi}와 $(g_X^{s_1 s_2 \cdots s_{i-1}})^{s_i} \bmod p$를 전송한다.

$$S_{Pi} = Sign_{pi}((g_X^{s_1 s_2 \cdots s_{i-1}})^{s_i} \bmod p \| X \| g_X)$$

단계 5) P_t는 P_{t-1}로 부터 받은 서명을 검증하고 다음과 같이 V_X를 계산한다.

$$V_X = g_X^{s_1 s_2 \cdots s_i} \bmod p$$

단계 6) P_t는 K'를 복원한다.

$$K' = T_X \oplus f(V_X)$$

단계 6) 복원한 비밀이 정확한지 알아보기 위해 f(K')를 계산하여 notice board에 공개된 f(K)값과 비교한다.

[비밀분산 프로토콜 (Ⅱ)]

단계 1) Dealer는 큰 두 소수 p, q의 곱으로 이루어진 N과 $(e, \varphi(N)) = 1$인 e를 선택하고 각 참가자의 비밀값 $S_i < N \ (1 \le i \le n)$ 들을 랜덤하게 선택한다.

단계 2) Dealer는 S_i값들을 비밀리에 각 참가자에게 전송한 후, 참가자의 ID와 함께 그 값들을 안전하게 저장한다.

단계 3) Dealer는 $X \in \Gamma^*$에 대해 다음과 같이 T_X값을 계산한다.

$$T_X = K \oplus f(\Pi_{x:P_x \subset X} S_x^{e'} \bmod N)$$

단계 4) Dealer는 각 Γ^*에 속하는 각 원소 X에 대해 (X, e, N, T_X), f(K) 값을 notice board에 공개한다.

[비밀복원 프로토콜(Ⅱ)]

단계 1) 각 참가자 P_i($P_i \in X$)는 X에 해당하는 N, e, T_X, f(K)를 notice board로부터 읽어온다.

단계 2) P_i는 $S_i^e \bmod N$ 계산하고 다음과 같이 디지털 서명을 생성한다.

$$S_{P_i} = Sig_{p_i}(S_i^e \bmod N \| X \| e \| N)$$

단계 3) 각 P_i는 P_j (단, $P_j \in X$이고 $P_j \neq P_j$) 에게 S_{p_i}와 $S_i^e \bmod N$를 전송한다.

단계 4) 각 P_i는 P_j로부터 받은 서명을 검증하고 다음과 같이 V_X를 계산한다.

$$V_X = \Pi_{x:Px \subset X} S_x^e \bmod N$$

단계 5) K'를 복원한다.

$$K' = T_X \oplus f(V_X)$$

단계 6) 복원한 비밀이 정확한지 알아보기 위해 f(K')를 계산하여 notice board에 공개된 f(K)값과 비교한다.

3. 제안하는 방식

본 장에서는 일방향 해쉬함수를 이용하여 기존의 온라인 비밀분산 방식들을 좀더 효율적으로 개선한 방식을 제안하고자 한다. 앞에서 설명한 Yeun의 방식에서는 각 참가자들이 RSA 암호방식 등을 이용하여 자신의 비밀정보를 드러내지 않고 다른 참가자들에게 전송하여 비밀을 복원하도록 하였다. 제안하는 방식에서는 이 과정에서 이산대수 문제나 RSA 암호방식 대신 일방향 해쉬함수를 사용함으로써 모듈라 멱승을 하는 기존의 방식에 비해 계산량 측면에서 좀더 효율적으로 개선하였다.

3.1 비밀 분산/복원 프로토콜

제안하는 방식은 비밀복원 과정에서 각 참여자의 비밀정보를 숨기기 위해 이산대수 문제나 RSA 암호방식 대신 일방향 해쉬함수를 사용하여 참가자에게 요구되는 계산량을 감소시켰다. 프로토콜은 다음과 같다.

[시스템 설정]

- P : 비밀분산에 참여하는 참가자 P_i $(1 \le i \le n)$의 집합
- D : dealer 단, $D \not\in P$
- Γ : access structure $(\Gamma \in 2^{|P|})$ X가 Γ의 원소일 경우, X에 속하는 참가자들의 비밀로부터 본래의 비밀 K를 복원할 수 있고, X가 Γ의 원소가 아닐 경우에는 비밀을 복원하는 것이 불가능하다
- Γ^* : Γ의 원소 중 비밀을 복원하는데 필요한 참가자의 수가 가장 적은 것들의 집합.
- K : 분산하고자 하는 비밀
- f, h : 충돌 회피성 일방향 해쉬함수
- S_{p_i} : 참가자 P_i의 디지털 서명

[비밀분산 프로토콜] (그림 3-1 참조)

단계 1) Dealer는 난수 r과 각 참가자의 비밀값 $S_i \in_R Z$ $(1 \le i \le n)$를 랜덤하게 선택한다.

단계 2) Dealer는 선택한 S_i값들을 비밀리에 각 참가자에게 전송한 후, 참가자의 ID와 함께 그 값들을 안전하게 저장한다.

단계 3) Dealer는 $X \in \Gamma^*$에 대해 다음과 같이 T_X값을 계산한다.

$$T_X = K \oplus f(h(S_1 \| r) \oplus h(S_2 \| r) \oplus \cdots \oplus h(S_n \| r))$$

단계 4) Dealer는 각 Γ^*에 속하는 각 원소 X에 대해 (X, r, T_X)와 f(K) 값을 notice board에 공개한다.

Dealer D		참가자 P_i
난수 r과 $S_i \in_R Z$ $(1 \le i \le n)$ 선택 ID와 $S_i (1 \le i \le n)$값 저장 $X \in \Gamma^*$에 대해 T_X 계산 $T_X = K \oplus f(h(S_1 \| r) \oplus h(S_2 \| r) \oplus \cdots \oplus h(S_n \| r))$ (X, r, T_X)와 f(K)값을 notice board에 공개	S_i (secret) ⟶	S_i값 저장

(그림 3-1) 제안하는 비밀분산 프로토콜

[비밀복원 프로토콜] (그림 3-2 참조)

단계 1) 각 참가자 P_i(단, $P_i \in X$)는 X에 해당하는 난수 r과 T_X, f(K)를 notice board로부터 읽어온다.

단계 2) P_i는 자신의 비밀정보 S_i와 난수 r을 이용하여 $h(S_i||r)$을 계산하고 다음과 같이 디지털 서명을 생성한다.

$$S_{P_i} = Sig_{p_i}(h(S_i||r)||X||r)$$

단계 3) 각 P_i는 P_j (단, $P_j \in X$이고 $P_j \neq P_j$)에게 S_{p_i}와 $h(S_i||r)$를 전송한다.

단계 4) 각 P_i는 P_j로부터 받은 서명을 검증하고 다음과 같이 V_X를 계산한다.

$$V_X = h(S_1||r) \oplus h(S_2||r) \oplus \cdots \oplus h(S_n||r)$$

단계 5) K'를 복원한다.

$$K' = T_X \oplus f(V_X)$$

단계 6) 복원한 비밀이 정확한지 알아보기 위해 f(K')를 계산하여 notice board에 공개된 f(K)값과 비교한다.

참가자 P_i		참가자 P_j ($i \neq j$)
r, T_X, f(K) $h(S_i\|\|r)$ 계산 서명 생성 $S_{P_i} = Sig_{p_i}(h(Si\|\|r)\|\|X\|\|r)$	$h(S_i\|\|r)$, S_{p_i} -------------->	서명을 검증하고 V_X를 계산 $V_X=h(S_1\|\|r) \oplus h(S_2\|\|r) \oplus \cdots \oplus h(S_n\|\|r)$ $K' = T_X \oplus f(V_X)$ $f(K') \doteq f(K)$ 확인

(그림 3-2) 제안하는 비밀복원 프로토콜

3.2 부정한 참가자의 구별

제안하는 방식은 부정한 참가자의 수에 상관없이 비밀복원 과정에서 올바른 비밀을 제공하지 않은 참가자들의 신분을 밝혀낼 수 있다. 복원된 K'의 해쉬값 f(K')가 f(K)와 다를 경우, 각 참가자들은 프로토콜의 진행과정 동안 받은 값들을 dealer에게 제출한다. Dealer는 각각의 서명을 검증한 후, 비밀분산 과정에서 저장해둔 S_i ($1 \le i \le n$)와 난수 r을 사용하여 $h(S_i||r)$ ($1 \le i \le n$)를 계산하고 참가자들이 제공한 값과 비교하여 부정한 참가자를 알아낼 수 있다.

3.3 다수의 비밀 분산으로의 확장

Dealer는 새로운 비밀 K_ℓ를 분산할 때마다 이전과 다른 access structure Γ를 구성하고 새로운 난수 r_ℓ를 선택하여 다음을 계산한다 (단, $\ell = 1, 2, ..., m$).

$$T_{x\ell} = K_\ell \oplus f(h(S_1||r_\ell) \oplus h(S_2||r_\ell) \oplus \cdots \oplus h(S_n||r_\ell))$$

그 후에, 각 비밀 K_ℓ에 대해 $(X, r_\ell, T_{x\ell})$와 $f(K_\ell)$ ($\ell = 1, 2, ..., m$)를 notice board에 공개한다. 이와 같은 방법으로 제안하는 방식은 각 참가자들이 갖고 있는 share S_i ($1 \le i \le n$)를 변경하지 않고 여러 개의 비밀을 분산하는데 이용할 수 있다.

4. 향후과제 및 결론

본 논문에서는 하나의 share만으로 여러 개의 비밀을 분산할 수 있는, 일방향 해쉬함수에 기반한 효율적인 온라인 비밀분산 방식을 제안하였다. 제안하는 방식은 액세스 구조가 변하는 경우에 notice board에 공개된 값들만 변경하면 각 참가자들은 기존의 share를 그대로 사용할 수 있으므로, 하나의 share만으로 여러 개의 비밀을 분산할 수 있다. 더욱이 참가자들의 부정이 있는 경우 그 수에 관계없이 부정한 참가자의 신분을 밝혀낼 수 있다.

제안하는 방식에서도 참가자의 서명생성에서는 여전히 공개키 암호방식이 요구된다. 이러한 문제점은 N.Asokan 등이 ESORICS'96에서 제안한 Server-supported signature (S^3) 서명 방식[1]을 이용하거나, 참가자와 dealer 사이의 MAC (Message Authentication Codes) 등을 이용하면 해결할 수 있다. 그러나 server-supported

signature 방식을 이용하는 경우에 참가자에게 서명 생성시 요구되는 계산량은 해쉬함수만으로 줄일 수 있으나, 참가자들의 서명을 생성하는 서버가 비밀 복원 과정에서 항상 온라인으로 연결되어 있어야 하는 단점이 있다. 또한 MAC을 사용하는 경우에는 dealer의 부정은 밝혀낼 수 없다는 문제가 있다.

향후에는 좀 더 효율적인, 완전히 일방향 해쉬함수에만 기반한, 부정한 참가자의 신분 확인이 가능한 온라인 비밀 분산 방식을 제안하고자 한다.

5. 참고문헌

[1] N. Asokan, G. Tsudik, M. Waidner, "Server-Supported Signature", In proceeding of ESORICS '96, Rome, Italy, Sept, pp. 25-27, 1996

[2] G. R. Blakely, "Safeguarding cryptographic key", In proceeding of AFIPS National Computer Conference, pp. 313-317, 1979

[3] C. Cachin, "On-line secret sharing", In C. Boyd, editor, Proceeding of the 5th IMA conference on Cryptography and Coding, pp. 190-198, Springer-Verlag, 1995

[4] W. Diffie and M. Hellman, "New Directions in Cryptography", IEEE Trransaction on Information Theory IT-22, pp. 644-654, 1976.

[5] H. Ghodosi, J. Pieprzyk, G. R. Chaudhry, J. Seberry, "How to prevent cheating in Pinch's scheme", Electronic Letters, 33(17) : 1453-1454, 1997

[6] A. Menezes, P. van Oorschot, S. Vanstone, "Handbook of Applied Cryptography", CRC Press, 1996

[7] R.G.E. Pinch, "On-line multiple secret sharing", Electronic Letters, 32(12) : 1087-1088, 1996

[8] R. Rivest, A. Shamir, and L. Adleman, "A method for Obtaining Digital signature and Public key Cryptosystems", Communication of the ACM, pp. 120-128, FEB. 1978.

[9] Adi Shamir, "How to share a secret", Communication of the ACM, 21 : 120-126, 1978

[10] M. Tompa, H. Woll, "How to share a secret with cheater", Journal of Cryptology, 1 : 133-138, 1988

[11] Chan Yeob Yeun, Chris J. Mitchell, "How to identify all cheater in Pinch's scheme", JWIS '98, Japan-Singapore Joint Workshop on Information Security, Singapore, Dec. 1998.

[12] Chan Yeob Yeun, Chris J. Mitchell, Mike Burmester, "An Online Secret Sharing which Identifies All Cheater", Proceeding of NORDSEC '98, The

Third Nordic Workshop on Secure IT Systems, Norway, Nov. 1998

[13] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game or a completeness theorem for protocols with honest majority", in Proc. 19th ACM Symposium on Theory of Computing (STOC), 1987, pp. 218-229.