<https://www.ibr.cs.tu-bs.de/papers/ruesch-chants2017.pdf>

**Forward Secure Delay-Tolerant Networking**

**[0. Forward Secure Encryption]**

|  |
| --- |
| **Forward Secure Encryption:** **키가 유출되었을 때 이전 메시지의 기밀성을 유지**하기 위해 **사용자의 secret key가 주기적으로 업데이트**된다는 것이다.   * **Puncturable Encryption의 아이디어:** 수신자는 새로운 키를 교환할 필요 없이 **복호화 키를 업데이트**하여 **지정된 메시지, 수신자 또는 time period에 대한 복호화 능력을 무효화**할 수 있다.   + 이때 time interval은 secret key의 각 부분에 매핑된다. * **Puncturable Encryption**과 **Forward Secure Public Key Encrpytion(FS-PKE) scheme**을 결합하여 구현할 수 있다. * 이 암호화 모델은 **송수신자가 보안되지 않은 채널을 통해 통신**한다고 가정한다. * **Asynchronous Communication에 초점**이 맞춰져 있는데, interactive에 대한 forward secrecy가 보장되면 synchronous communication은 다른 프로토콜에 의해 보장되기 때문이다. |

**[1. Puncturable Encryption]**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Puncturable Encryption:** tag-based 암호화 알고리즘의 한 형태로, 짧고 바뀌지 않는 public key를 사용하고, puncture 알고리즘을 포함한다.   * **Puncturing:** **현재 secret key SK와** **태그**  에 대하여, **t로 암호화되지 않은 모든 암호문을 복호화하는 새로운 비밀키 SK’**를 만든다. (Attribute-Based Encryption 기반) * Active attacker는 메시지를 차단하여 수신자가 복호화할 수 없게 만들 수 있지만, **두 scheme이 결합**된, **tag와 time interval에 대한 revocation**이 가능한 알고리즘을 사용하면 **어떤 decryption window를 지난 메시지는 복호화될 수 없다**는 것을 보장할 수 있다.   + 이 알고리즘, 즉 **Forward Secure Encryption (FSE)**은 다음과 같다.  |  |  | | --- | --- | |  | Security parameter d와 tag의 최대 개수 k가 주어졌을 때 **새로운 public key와 initial secret key**를 반환한다. | |  |  | |  |  | |  | [2. Puncturing of Keys] | |  | [3. Key Forwarding] |  * Key pair가 생성되는 동안…   + ABE scheme의 instance에 대한 parameter가 생성된다.   + public key에 의해 public parameter들이 생성된다.   + Master secret key에 의해 복호화 키가 생성되고, master secret key는 파괴된다. |

**[2. Puncturing of Keys]**

|  |
| --- |
| 일반적인 puncturing 알고리즘은 오른쪽 그림과 같다.   1. 각 암호문에 대해 **sender에 의해 생성된 tag가 추가**된다. 2. 사용자가 tag된 암호문을 받으면, **secret key 업데이트를 통해 secret key의 해당 tag에 대한 복호화 능력을 소멸**시킨다. **(puncturing on that tag)**  * 한 암호문에 여러 개의 태그를 사용할 수 있다. |

**[3. Key Forwarding]**

|  |
| --- |
| **Key Forwarding:** FSE 키의 lifetime은 여러 개의, steady한 서로 다른 길이의 time interval로 나뉘어지는데, 이 **time interval로부터 새로운 secret key를 얻는** 과정   * 오래된 키는 **기각**되거나 **late interval case로 지정**될 수 있다. 또는 **복호화는 가능하지만 새로운 키를 얻기 위해서 사용할 수 없게** 될 수 있다. * Tag는 **다음 interval에 대한 키가 이미 얻어졌을 때만 puncture**될 수 있다. * Interval key가 해독될 수 있으므로 키를 이전 interval에 무기한 저장하는 것도 권장되지 않는다. |

**[4. Introduction of Forward Secure DTNs]**

|  |
| --- |
| **FSE의 구조를 DTN에 적용**한다.   1. 새로운 scheme의 조합을 **Bundle Security Protocol에 적용**하는 것   [5. Extending the Bundle Protocol]   1. Bundle Protocol specification과 호환되는 **Unique tag를 만드는** 방법   [6. Unique Tags in DTNs]   1. **중요한 parameter**들 (FSE는 DTN 시나리오에 맞게 설정되어야 하므로)   [7. FSE Parameters]   1. **Implementation-specific**한 고려 사항들   [8. Implementation] |

**[5. Extending the Bundle Protocol]**

|  |
| --- |
| **목적:** **송신자와 수신자 사이에서 전송되는 모든 bundle**에 대해, Bundle Security Protocol에 의해 제공되는 보안을 **Forward secrecy를 이용하여 확장**한다.  **기본 사항:** FSE scheme [0. Forward Secure Encryption] [1. Punturable Encryption] [2. Puncturing of Keys] [3. Key Forwarding] [4. Introduction of Forward Secure DTNs] 을 이용하면 **각 bundle에는 unique tag가 포함**되고, bundle을 수신하면 **수신자의 secret key는 이 tag에 puncture**되고, 이로 인해 **복호화를 할 수 없게** 된다.   * FSE scheme은 이미 정의된 bundle type에 대한 추가적인 수정 없이 **Bundle Security Protocol의 일반적인 ciphersuite에 대한 alternative**로 추가된다. * Key management는 변경되지 않고, RSA key 대신 **FSE public key**가 교환된다. |
| **ciphersuite(암호화 스위트):** 전송 계층 보안(TLS)에 쓰이는 인터넷 보안 보조 알고리즘으로, 구조는 다음과 같다.   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | TLS | ECDHE | RSA | AES\_128 | CBC | SHA | | 프로토콜 | 키 교환방식 | 인증서 검증 | 대칭키 블록 암호화 방식 | 블록 암호 운용 방식 | 메시지 인증  (무결성) |   (<https://ko.wikipedia.org/wiki/%EC%95%94%ED%98%B8%ED%99%94_%EC%8A%A4%EC%9C%84%ED%8A%B8>) |

**[6. Unique Tags in DTNs]**

|  |
| --- |
| FSE scheme에서 **각 메시지는 1개 또는 여러 개의 tag**를 가지고 있다.   * 각 tag는 **메시지를 unique하게 identify**하거나 **subject에 따라서 묶는** 용도 등으로 사용된다. * Generic DTN 통신에서는 모든 bundle이 **tag를 이용하여 uniquely identifiable**할 수 있고, **수신자에 의해 복호화**될 수 있다고 가정한다.   + 따라서 **timestamp 값**과 **노드의 EID의 해시 값**을 이용하여 tag를 생성하는 것을 제안한다. (예: for node A, t=h(“A-2017-06-04”) and k=1)   (참고: [0. Forward Secure Encryption] [1. Punturable Encryption] [2. Puncturing of Keys] [3. Key Forwarding] [4. Introduction of Forward Secure DTNs])   * + 그 직후 이 bundle에 대한 복호화는 **이 tag의 FSE key의 puncturing에 의해 영구적으로 실행할 수 없어야** 한다. * 각 bundle은 그것들의 **source endpoint, creation timestamp, creation timestamp sequence number에 대해 unique**할 것이라고 가정한다.   + 따라서 이 값들의, **bundle의 unique tag로 쓰이는 해시 값을 계산**할 수 있고, 이를 통해 **각 bundle에 대해 forward secrecy를 제공**할 수 있다. |

**[7. FSE Paramters]**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| FSE의 parameter에는 **interval length, amount of tags, amount of time intervals, how many current valid interval keys are stored**가 있다.   |  |  | | --- | --- | | Interval length **(n)** | 여러 개의, steady한 서로 다른 길이의 **Time interval로 나뉘어진 FSE key의 lifetime**   * 암호화 및 복호화 성능의 최적화를 위해 interval 당 1개의 메시지를 수신하는 것이 좋다. 즉, **오직 1번의 puncturing 연산을 수행하고 즉시 key를 forward**한다. | | Amount of tags **(k)** | 메시지에는 **amount of tag의 값 이 포함**될 수 있다.   * Libforwardsec의 기본 설정은 메시지 당 1개의 태그를 포함하는 것이다. | | Amount of time intervals **(d)** | **d time interval 이후에 새로운 FSE key들이 생성 및 교환**되어야 한다는 것을 의미한다.   * FSE key는 특정한 amount of time intervals를 지원한다. * Libforwardsec에 의해 최대 개의 time interval이 지원된다. * **key들이 주기적으로 forward**되고 따라서 **지난 interval의 메시지를 복호화하는 능력을 잃게** 되므로, 오래된 interval key는 메시지의 **early/late arrival을 지원**하도록 해야 한다. | | How many currently valid internal keys are stored **(N)** | **현재 유효한 키가 얼마나 많이 저장되어 있는지**를 의미한다.   * 전송되었으나 수신되지 않은 **lost bundle**이 있는 상황에서 도움이 된다.   + Lost bundle은 특정한 time interval에서 암호화된다. * Secret key가 무기한 저장되지 않으므로, **대응되는 secret key는 N time interval이 지난 후 폐기**된다. 따라서 이 bundle들에 대한 **forward secrecy 역시 보장**된다. * 특별한 시나리오에서는 **key들이 사용되기 전에 미리 유도**될 수 있다. |  * 서로 다른 종류의 네트워크에서 이 값들에 대한 최적의 선택은 **주어진 환경과 선택된 application에 크게 좌우**된다. |
| **Libforwardsec:** <https://github.com/imichaelmiers/libforwardsec>  “Forward secure encryption for asynchronous messaging”   * Clock skew, delayed 또는 lost message 문제에 대응할 수 있는, 효과적인 forward-secure 퍼블릭 키 암호화를 제공한다. * 이것을 이용하여 forward secure O-RTT 키 교환을 빌드하거나 오프라인 상태에서 forward secure message를 전송할 수 있다. |

**[8. Implementation]**

|  |
| --- |
| Forward secrecy를 제공하는 이 암호화 scheme은 **libforwardsec 라이브러리에 구현**되어 있다.   * Libforwardsec의 설정값은 **256비트 Barreto-Naehrig curve를 이용**하고, **symmetric curve를 asymmetric curve로 변형한 것에서 최소 암호문 크기를 최적화**한다.   + 이것의 기반 가정은 secret key storage에는 대부분의 DTN 환경에서 사용 가능한 높은 제약 조건의 device 외에는 **아무 문제점이 없다**는 것이다. |

**[9. Evaluation]**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **벤치마크:** Dell OptiPlex 7010 Desktop-PC with an Intel Core i7-3770 CPU @ 4(8)x3.4 GHz CPU and 16GB of RAM, running Ubuntu 14.04 LTS   |  |  | | --- | --- | | **Figure 3a**  Key generation  **Figure 3b**  Start-up time of Security-KeyManager |  | | **Figure 4**  Encryption/Decryption |  | | **Figure 5a**  Latency introduced by FSE  **Figure 5b**  Latency during interval progression |  | |