Normas, Investigación y Aplicaciones en Criptografía en NIST

Algunas notas sobre PEC y PQC*

Luís Brandão y René Peralta

Cryptographic Technology Group
National Institute of Standards and Technology
(Gaithersburg, Maryland, USA)

Presentación en la Universidad de Chile
7 de Octubre, 2019 @ Santiago, Chile

* Presented in Spanish (Castellano) per preference of the host. English version is available upon request. Some slides are based on previous presentations (ACS'19, ZKProof'19, NuTMiC'17, PETS'15). Minor corrections on October 14, 2019.

Pauta

1. Introducción — Cripto en NIST
2. Criptografía de mejora de la privacidad (PEC)
3. Criptografía post-cuántica (PQC)
4. Observaciones finales

Pauta

1. Introducción — Cripto en NIST
2. Criptografía de mejora de la privacidad (PEC)
3. Criptografía post-cuántica (PQC)
4. Observaciones finales

Objetivos de esta presentación

- Difusión de proyectos interesantes / áreas de investigación.
- Motivar la colaboración con NIST (aplicaciones PEC)

Pauta 1

Algunos datos de NIST

Instituto Nacional de Normalización y Tecnología (NIST)

(National Bureau of Standards 1901–1988 → NIST 1988–present)

- Agencia federal (en el Departamento de Comercio de EE. UU.), no reguladora
- **Misión** (palabras clave): innovación, competitividad industrial, ciencia, normas y metrología, seguridad económica, calidad de vida.

Foto aérea del campus en Gaithersburg (fuente: Gogle Maps, Agosto, 2019)

Algunos datos de NIST

Instituto Nacional de Normalización y Tecnología (NIST)

(National Bureau of Standards 1901–1988 → NIST 1988–present)

- Agencia federal (en el Departamento de Comercio de EE. UU.), no reguladora
- **Misión** (palabras clave): innovación, competitividad industrial, ciencia, normas y metrología, seguridad económica, calidad de vida.

Amplia gama de competencias

- Personal de ~ 6–7×10^3
- Cinco laboratorios y dos centros
- Laboratorios → Divisiones → Grupos → Proyectos
- Normas, investigación y aplicaciones

Foto aérea del campus en Gaithersburg (fuente: Gogle Maps, Agosto, 2019)

Laboratorios, divisiones, grupos

Laboratorio de Tecnologías de la Información (ITL):

Investigación y desarrollo en metrología, estándares, y tecnología en las áreas de informática, matemáticas, y estadística.

Laboratorios, divisiones, grupos

Laboratorio de Tecnologías de la Información (ITL):

Investigación y desarrollo en metrología, estándares, y tecnología en las áreas de informática, matemáticas, y estadística.

→ **División de Seguridad Computacional (CSD):** Tecnologías Criptográficas; Sistemas Seguros y Aplicaciones; Componentes Seguros y Mecanismos; Ingeniería de Seguridad y Gestión de Riesgos; Pruebas, Validación, y Metrología de Seguridad.

Laboratorios, divisiones, grupos

Laboratorio de Tecnologías de la Información (ITL):

Investigación y desarrollo en metrología, estándares, y tecnología en las áreas de informática, matemáticas, y estadística.

- → **División de Seguridad Computacional (CSD):** Tecnologías Criptográficas; Sistemas Seguros y Aplicaciones; Componentes Seguros y Mecanismos; Ingeniería de Seguridad y Gestión de Riesgos; Pruebas, Validación, y Metrología de Seguridad.
  - → **Grupo de Tecnologías Criptográficas (CTG):** investigación y desarrollo, pautas, recomendaciones, buenas prácticas en algoritmos, métodos, y protocolos criptográficos.

Laboratorios, divisiones, grupos

Laboratorio de Tecnologías de la Información (ITL):

Investigación y desarrollo en metrología, estándares, y tecnología en las áreas de informática, matemáticas, y estadística.

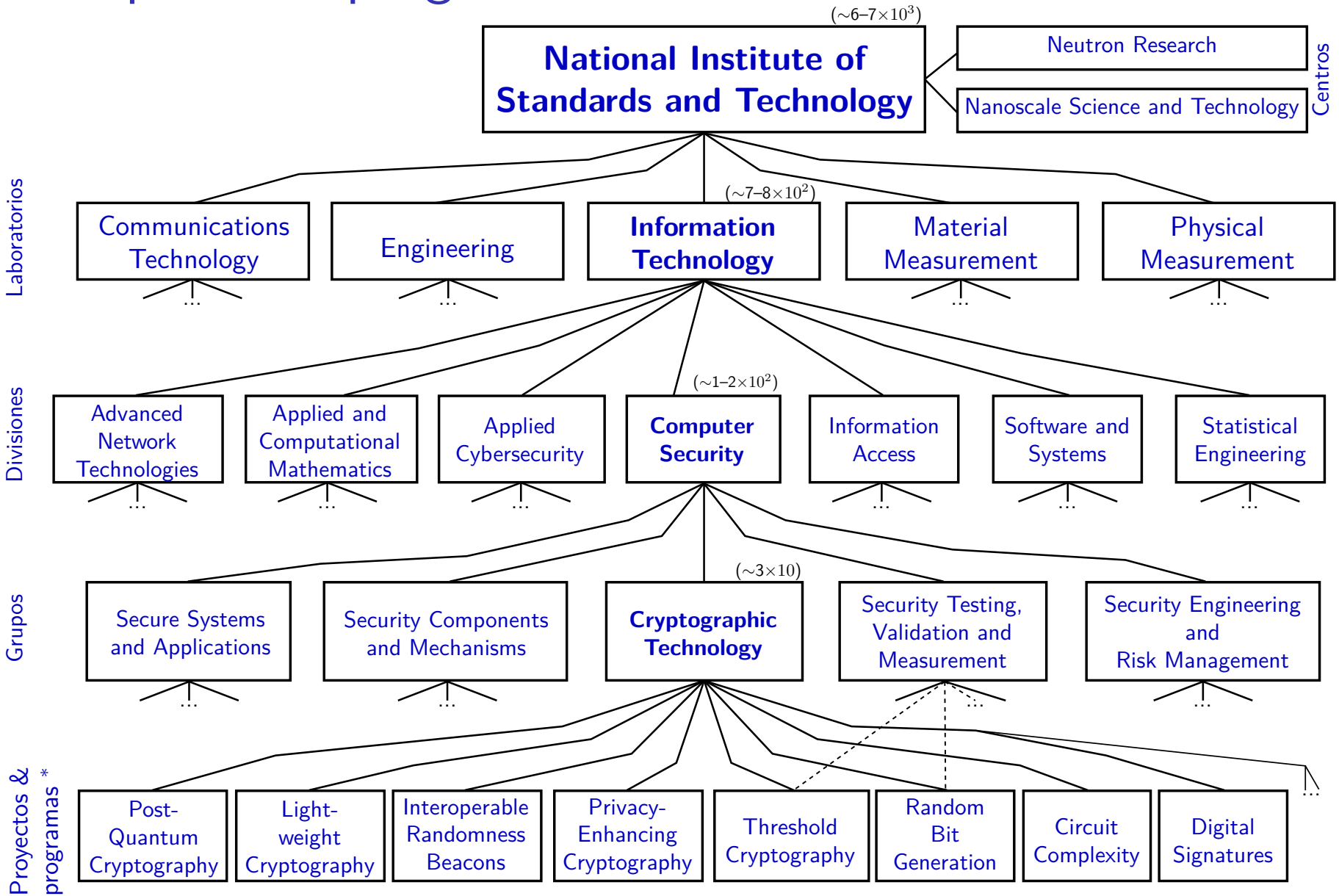
→ **División de Seguridad Computacional (CSD):** Tecnologías Criptográficas; Sistemas Seguros y Aplicaciones; Componentes Seguros y Mecanismos; Ingeniería de Seguridad y Gestión de Riesgos; Pruebas, Validación, y Metrología de Seguridad.

→ **Grupo de Tecnologías Criptográficas (CTG):** investigación y desarrollo, pautas, recomendaciones, buenas prácticas en algoritmos, métodos, y protocolos criptográficos.

- Documentos: FIPS, SP 800, NISTIR.
- Cooperación internacional: gobierno, industria, mundo académico, organismos de normalización.

FIPS = Federal Information Processing Standards; SP 800 = Special Publications in Computer Security; NISTIR = NIST Internal or Interagency Report.

El Grupo de Criptografía en NIST



* (Algunos proyectos / programas involucran a varios grupos, divisiones o labs.)

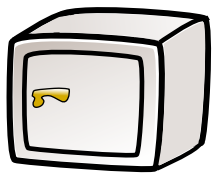
(entre paréntesis: # aproximado de empleados federales y associados)

Algunos conceptos útiles para esta charla

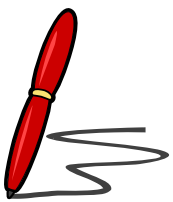
- **Texto encriptado / cifra:**

- **Hash:**

- **Compromiso (Commitment):**



- **Firma [digital] (Signature):**

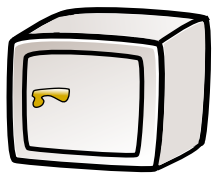


Algunos conceptos útiles para esta charla

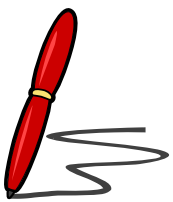
- **Texto encriptado / cifra:**
  - como un texto revuelto, pareciendo aleatorio
  - necesita una clave para ser leída

- **Hash:**

- **Compromiso (Commitment):**



- **Firma [digital] (Signature):**

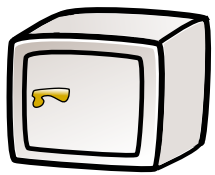


Algunos conceptos útiles para esta charla

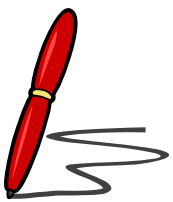
- **Texto encriptado / cifra:**
  - como un texto revuelto, pareciendo aleatorio
  - necesita una clave para ser leída

- **Hash:**
  - como una huella dactilar de datos (secuencia "única"de 512 bits)
  - parece aleatorio si se desconocen sus datos de origen

- **Compromiso (Commitment):**



- **Firma [digital] (Signature):**

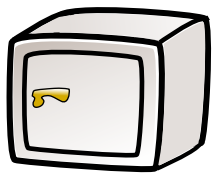


Algunos conceptos útiles para esta charla

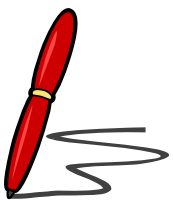
- **Texto encriptado / cifra:**
  - como un texto revuelto, pareciendo aleatorio
  - necesita una clave para ser leída

- **Hash:**
  - como una huella dactilar de datos (secuencia "única"de 512 bits)
  - parece aleatorio si se desconocen sus datos de origen

- **Compromiso (Commitment):**
  - como una bóveda que oculta datos, hasta que se abre
  - una vez cerrado, no se puede cambiar lo que hay dentro



- **Firma [digital] (Signature):**

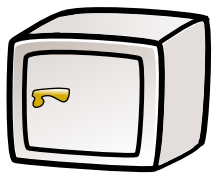


Algunos conceptos útiles para esta charla

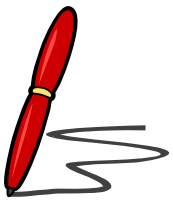
- **Texto encriptado / cifra:**
  - como un texto revuelto, pareciendo aleatorio
  - necesita una clave para ser leída

- **Hash:**
  - como una huella dactilar de datos (secuencia "única"de 512 bits)
  - parece aleatorio si se desconocen sus datos de origen

- **Compromiso (Commitment):**
  - como una bóveda que oculta datos, hasta que se abre
  - una vez cerrado, no se puede cambiar lo que hay dentro



- **Firma [digital] (Signature):**
  - como una firma física, pero no se puede falsificar
  - la firma copiada no es válida para otro documento



Algunas primitivas criptográficas normalizadas

Núcleo tradicional:

Algunas primitivas criptográficas normalizadas

Núcleo tradicional:

- Cifrado por bloques
- Modos de uso
- Funciones de hash
- Firmas digitales
- Acuerdo de claves
- DRBGs

Algunas primitivas criptográficas normalizadas

Núcleo tradicional:

- Cifrado por bloques: DES (1977), EES (1994), TDEA (1999), AES (2001)
- Modos de uso (1980–): CBC, CT, CCM, GCM ...
- Funciones de hash (SHS): SHA-1 (1994), SHA-2 (2001), SHA-3 (2015)
- Firmas digitales (DSS): DSA (1997), ECDSA (1998), RSA (2000), EdDSA (2019)
- Acuerdo de claves, e.g., based on DH (2006) and RSA (2009)
- DRBGs (2006): CTR_, Hash_, HMAC_, Dual_EC_
  (withdrawn in 2015 due to concerns of potential subversion)

(Lista no exhaustiva; años indicados para perspectiva; alguna documentación tiene actualizaciones posteriores)
(Más detalles en "NIST Cryptographic Standards and Guidelines Development Program Briefing Book")

Leyenda:
- AES = Advanced Encryption Standard
- CBC = Cipher block chaining (mode)
- CT = Counter (mode)
- CCM = Counter with Cipher-block chaining
- DES = Data Encryption Standard
- DH = Diffie–Hellman
- DSA = Digital Signature Algorithm
- DSS = Digital Signature Standard
- DRBG = Deterministic Random Bit Generator
- ECDSA = Elliptic curve DSA
- EdDSA = Edwards curve DSA
- EES = Escrowed Encryption Standard
- GCM = Galois counter mode
- RSA = Rivest–Shamir–Adleman
- SHA = Secure Hash Algorithm
- SHS = Secure Hash Standard
- TDEA = Triple Data Encryption Algorithm

Algunas primitivas criptográficas normalizadas

Núcleo tradicional:

- Cifrado por bloques: DES (1977), EES (1994), TDEA (1999), AES (2001)
- Modos de uso (1980–): CBC, CT, CCM, GCM ...
- Funciones de hash (SHS): SHA-1 (1994), SHA-2 (2001), SHA-3 (2015)
- Firmas digitales (DSS): DSA (1997), ECDSA (1998), RSA (2000), EdDSA (2019)
- Acuerdo de claves, e.g., based on DH (2006) and RSA (2009)
- DRBGs (2006): CTR_, Hash_, HMAC_, Dual_EC_
  (withdrawn in 2015 due to concerns of potential subversion)

(Lista no exhaustiva; años indicados para perspectiva; alguna documentación tiene actualizaciones posteriores)
(Más detalles en "NIST Cryptographic Standards and Guidelines Development Program Briefing Book")

Leyenda:
- AES = Advanced Encryption Standard
- CBC = Cipher block chaining (mode)
- CT = Counter (mode)
- CCM = Counter with Cipher-block chaining
- DES = Data Encryption Standard
- DH = Diffie–Hellman
- DSA = Digital Signature Algorithm
- DSS = Digital Signature Standard
- DRBG = Deterministic Random Bit Generator
- ECDSA = Elliptic curve DSA
- EdDSA = Edwards curve DSA
- EES = Escrowed Encryption Standard
- GCM = Galois counter mode
- RSA = Rivest–Shamir–Adleman
- SHA = Secure Hash Algorithm
- SHS = Secure Hash Standard
- TDEA = Triple Data Encryption Algorithm

Algunas primitivas criptográficas normalizadas

Núcleo tradicional:

- Cifrado por bloques: DES (1977), EES (1994), TDEA (1999), AES (2001)
- Modos de uso (1980–): CBC, CT, CCM, GCM ...
- Funciones de hash (SHS): SHA-1 (1994), SHA-2 (2001), SHA-3 (2015)
- Firmas digitales (DSS): DSA (1997), ECDSA (1998), RSA (2000), EdDSA (2019)
- Acuerdo de claves, e.g., based on DH (2006) and RSA (2009)
- DRBGs (2006): CTR_, Hash_, HMAC_, Dual_EC_
  (withdrawn in 2015 due to concerns of potential subversion)

(Lista no exhaustiva; años indicados para perspectiva; alguna documentación tiene actualizaciones posteriores)
(Más detalles en "NIST Cryptographic Standards and Guidelines Development Program Briefing Book")

Leyenda:
- AES = Advanced Encryption Standard
- CBC = Cipher block chaining (mode)
- CT = Counter (mode)
- CCM = Counter with Cipher-block chaining
- DES = Data Encryption Standard
- DH = Diffie–Hellman
- DSA = Digital Signature Algorithm
- DSS = Digital Signature Standard
- DRBG = Deterministic Random Bit Generator
- ECDSA = Elliptic curve DSA
- EdDSA = Edwards curve DSA
- EES = Escrowed Encryption Standard
- GCM = Galois counter mode
- RSA = Rivest–Shamir–Adleman
- SHA = Secure Hash Algorithm
- SHS = Secure Hash Standard
- TDEA = Triple Data Encryption Algorithm

Algunas primitivas criptográficas normalizadas

Núcleo tradicional:

- Cifrado por bloques: DES (1977), EES (1994), TDEA (1999), AES (2001)
- Modos de uso (1980–): CBC, CT, CCM, GCM ...
- Funciones de hash (SHS): SHA-1 (1994), SHA-2 (2001), SHA-3 (2015)
- Firmas digitales (DSS): DSA (1997), ECDSA (1998), RSA (2000), EdDSA (2019)
- Acuerdo de claves, e.g., based on DH (2006) and RSA (2009)
- DRBGs (2006): CTR_, Hash_, HMAC_, Dual_EC_
  (withdrawn in 2015 due to concerns of potential subversion)

(Lista no exhaustiva; años indicados para perspectiva; alguna documentación tiene actualizaciones posteriores)
(Más detalles en "NIST Cryptographic Standards and Guidelines Development Program Briefing Book")

Leyenda:
- AES = Advanced Encryption Standard
- CBC = Cipher block chaining (mode)
- CT = Counter (mode)
- CCM = Counter with Cipher-block chaining
- DES = Data Encryption Standard
- DH = Diffie–Hellman
- DSA = Digital Signature Algorithm
- DSS = Digital Signature Standard
- DRBG = Deterministic Random Bit Generator
- ECDSA = Elliptic curve DSA
- EdDSA = Edwards curve DSA
- EES = Escrowed Encryption Standard
- GCM = Galois counter mode
- RSA = Rivest–Shamir–Adleman
- SHA = Secure Hash Algorithm
- SHS = Secure Hash Standard
- TDEA = Triple Data Encryption Algorithm

Algunas primitivas criptográficas normalizadas

Núcleo tradicional:

- Cifrado por bloques: DES (1977), EES (1994), TDEA (1999), AES (2001)
- Modos de uso (1980–): CBC, CT, CCM, GCM ...
- Funciones de hash (SHS): SHA-1 (1994), SHA-2 (2001), SHA-3 (2015)
- Firmas digitales (DSS): DSA (1997), ECDSA (1998), RSA (2000), EdDSA (2019)
- Acuerdo de claves, e.g., based on DH (2006) and RSA (2009)
- DRBGs (2006): CTR_, Hash_, HMAC_, Dual_EC_
  (withdrawn in 2015 due to concerns of potential subversion)

(Lista no exhaustiva; años indicados para perspectiva; alguna documentación tiene actualizaciones posteriores)
(Más detalles en "NIST Cryptographic Standards and Guidelines Development Program Briefing Book")

Algunas de estas normas se especificaron con referencia a normas de otros organismos, y con requisitos adicionales.

Leyenda:
- AES = Advanced Encryption Standard
- CBC = Cipher block chaining (mode)
- CT = Counter (mode)
- CCM = Counter with Cipher-block chaining
- DES = Data Encryption Standard
- DH = Diffie–Hellman
- DSA = Digital Signature Algorithm
- DSS = Digital Signature Standard
- DRBG = Deterministic Random Bit Generator
- ECDSA = Elliptic curve DSA
- EdDSA = Edwards curve DSA
- EES = Escrowed Encryption Standard
- GCM = Galois counter mode
- RSA = Rivest–Shamir–Adleman
- SHA = Secure Hash Algorithm
- SHS = Secure Hash Standard
- TDEA = Triple Data Encryption Algorithm

Algunas primitivas criptográficas normalizadas

Núcleo tradicional:

- Cifrado por bloques: DES (1977), EES (1994), TDEA (1999), AES (2001)
- Modos de uso (1980–): CBC, CT, CCM, GCM ...
- Funciones de hash (SHS): SHA-1 (1994), SHA-2 (2001), SHA-3 (2015)
- Firmas digitales (DSS): DSA (1997), ECDSA (1998), RSA (2000), EdDSA (2019)
- Acuerdo de claves, e.g., based on DH (2006) and RSA (2009)
- DRBGs (2006): CTR_, Hash_, HMAC_, Dual_EC_
  (withdrawn in 2015 due to concerns of potential subversion)

(Lista no exhaustiva; años indicados para perspectiva; alguna documentación tiene actualizaciones posteriores)
(Más detalles en "NIST Cryptographic Standards and Guidelines Development Program Briefing Book")

Algunas de estas normas se especificaron con referencia a normas de otros organismos, y con requisitos adicionales.

Varios métodos:

- Técnicas desarrolladas en casa
- Adopción de normas externas
- Llamada abierta, competencia, otros procesos de consenso internacional

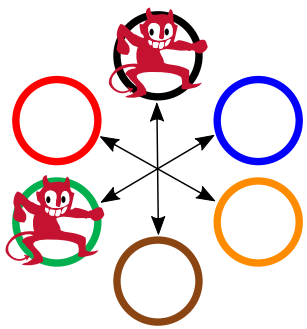
Leyenda:
- AES = Advanced Encryption Standard
- CBC = Cipher block chaining (mode)
- CT = Counter (mode)
- CCM = Counter with Cipher-block chaining
- DES = Data Encryption Standard
- DH = Diffie–Hellman
- DSA = Digital Signature Algorithm
- DSS = Digital Signature Standard
- DRBG = Deterministic Random Bit Generator
- ECDSA = Elliptic curve DSA
- EdDSA = Edwards curve DSA
- EES = Escrowed Encryption Standard
- GCM = Galois counter mode
- RSA = Rivest–Shamir–Adleman
- SHA = Secure Hash Algorithm
- SHS = Secure Hash Standard
- TDEA = Triple Data Encryption Algorithm

Criptografía de umbral

Proyecto NIST

Propósito: normalizar primitivas criptografícas con umbral
(firmas digitales, desencriptar*, generación de claves, cifrar/decifrar, ...)

* dentro de un esquema de cifrado de clave pública (PKE)



https://csrc.nist.gov/Projects/Threshold-Cryptography

Criptografía de umbral

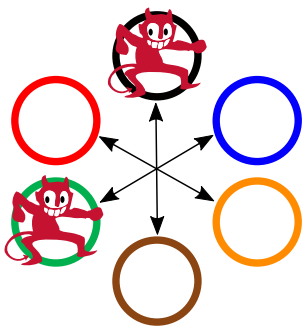
Proyecto NIST

Propósito: normalizar primitivas criptografícas con umbral
(firmas digitales, desencriptar*, generación de claves, cifrar/decifrar, ...)

* dentro de un esquema de cifrado de clave pública (PKE)

Algunas Propiedades:

- Tolerancia a la corrupción de f-de-n componentes
- Cada componente opera solo con una *parte* de la clave
- Mejor resistencia a los ataques de canal lateral



https://csrc.nist.gov/Projects/Threshold-Cryptography

Criptografía de umbral

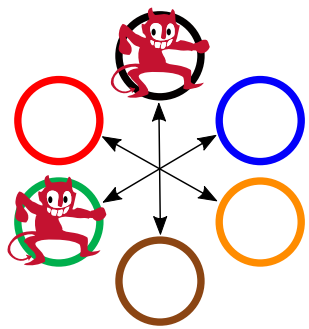
Proyecto NIST

Propósito: normalizar primitivas criptografícas con umbral
(firmas digitales, desencriptar*, generación de claves, cifrar/decifrar, ...)

* dentro de un esquema de cifrado de clave pública (PKE)

Algunas Propiedades:

- Tolerancia a la corrupción de f-de-n componentes
- Cada componente opera solo con una *parte* de la clave
- Mejor resistencia a los ataques de canal lateral



Pasos:

- (Marzo de 2019) **NISTIR 8214:** NIST Informe sobre Esquemas de Umbral
- (Marzo de 2019) **NTCW 2019:** NIST Taller de Criptografía de Umbral
- (Luego) **Plan preliminar de normalización**
- (Después) Consultar partes interesadas; desarrollar criterios para contribuciones

https://csrc.nist.gov/Projects/Threshold-Cryptography

Faros de aleatoriedad interoperables

Proyecto NIST

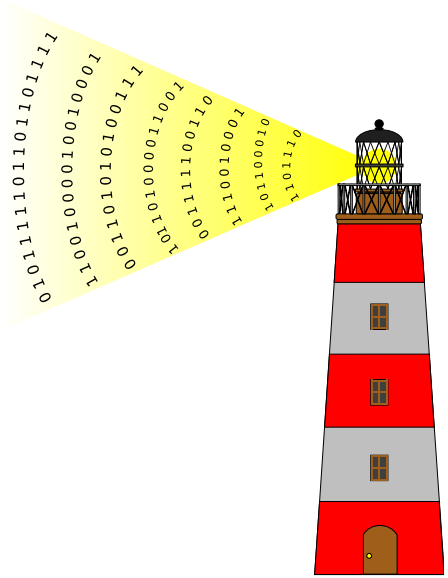
https://csrc.nist.gov/Projects/Interoperable-Randomness-Beacons

Faros de aleatoriedad interoperables

Proyecto NIST

Un Faro de aliatoriedad — a grandes rasgos:

- Un servicio que produce aleatoriedad pública en forma periódica



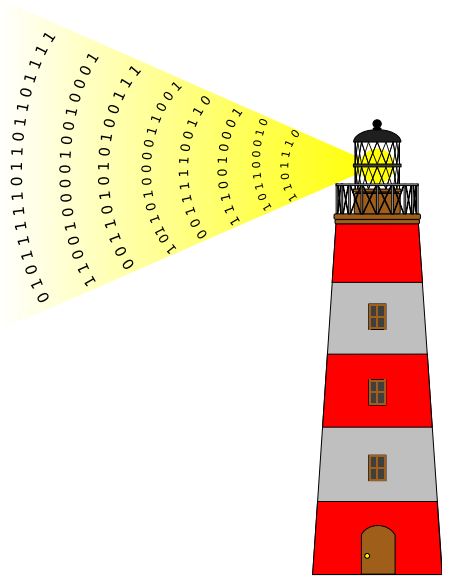
https://csrc.nist.gov/Projects/Interoperable-Randomness-Beacons

Faros de aleatoriedad interoperables

Proyecto NIST

Un Faro de aliatoriedad — a grandes rasgos:

- Un servicio que produce aleatoriedad pública en forma periódica
- *Pulso* periódico de aleatoriedad



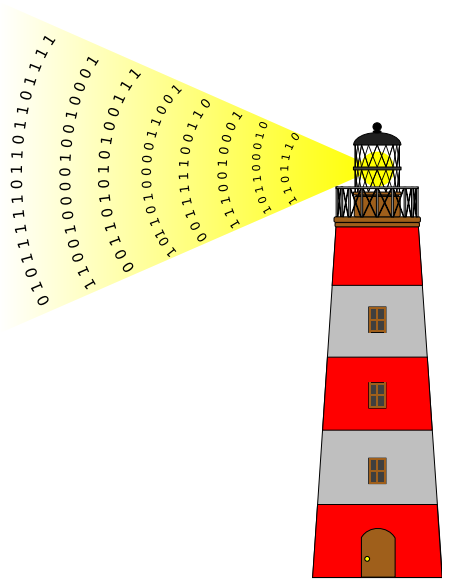
https://csrc.nist.gov/Projects/Interoperable-Randomness-Beacons

Faros de aleatoriedad interoperables

Proyecto NIST

Un Faro de aliatoriedad — a grandes rasgos:

- Un servicio que produce aleatoriedad pública en forma periódica
- *Pulso* periódico de aleatoriedad
- Cada pulso contiene una secuencia de 512 bits aleatorios recién generados



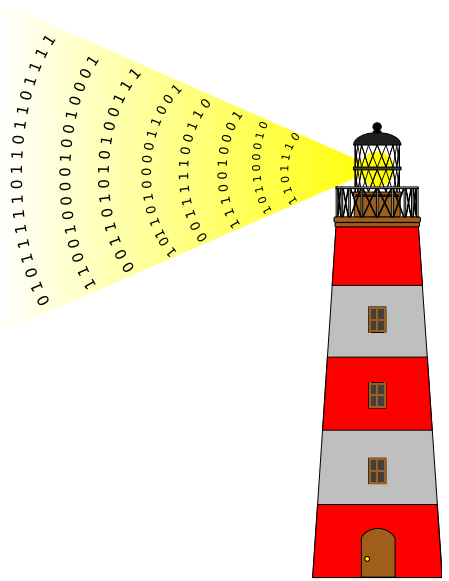
https://csrc.nist.gov/Projects/Interoperable-Randomness-Beacons

Faros de aleatoriedad interoperables

Proyecto NIST

Un Faro de aliatoriedad — a grandes rasgos:

- Un servicio que produce aleatoriedad pública en forma periódica
- *Pulso* periódico de aleatoriedad
- Cada pulso contiene una secuencia de 512 bits aleatorios recién generados
- Los pulsos están indexados, tienen un sello de tiempo y una firma digital



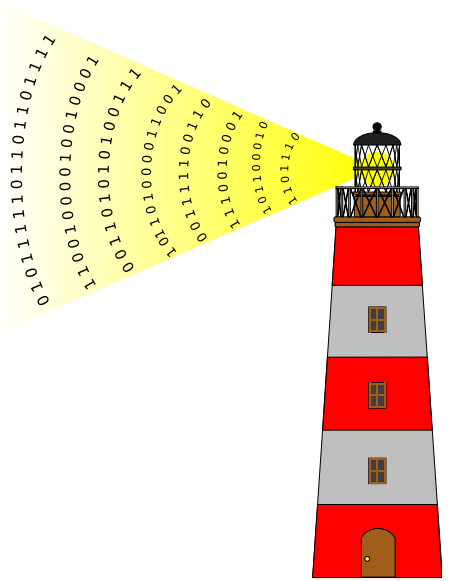
https://csrc.nist.gov/Projects/Interoperable-Randomness-Beacons

Faros de aleatoriedad interoperables

Proyecto NIST

Un Faro de aliatoriedad — a grandes rasgos:

- Un servicio que produce aleatoriedad pública en forma periódica
- *Pulso* periódico de aleatoriedad
- Cada pulso contiene una secuencia de 512 bits aleatorios recién generados
- Los pulsos están indexados, tienen un sello de tiempo y una firma digital
- Pulsos pasados son de acceso público



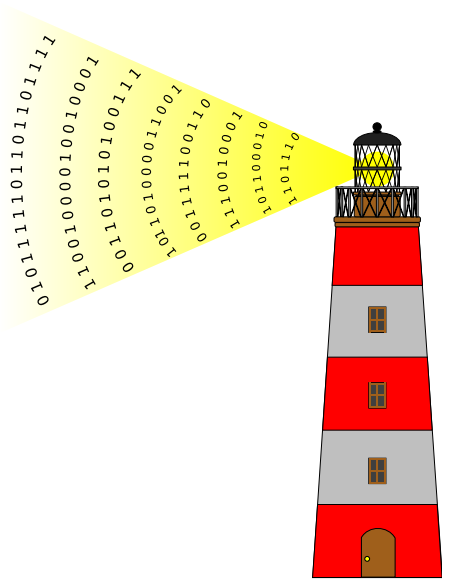
https://csrc.nist.gov/Projects/Interoperable-Randomness-Beacons

Faros de aleatoriedad interoperables

Proyecto NIST

Un Faro de aliatoriedad — a grandes rasgos:

- Un servicio que produce aleatoriedad pública en forma periódica
- *Pulso* periódico de aleatoriedad
- Cada pulso contiene una secuencia de 512 bits aleatorios recién generados
- Los pulsos están indexados, tienen un sello de tiempo y una firma digital
- Pulsos pasados son de acceso público
- La secuencia de pulsos forma una cadena de hash



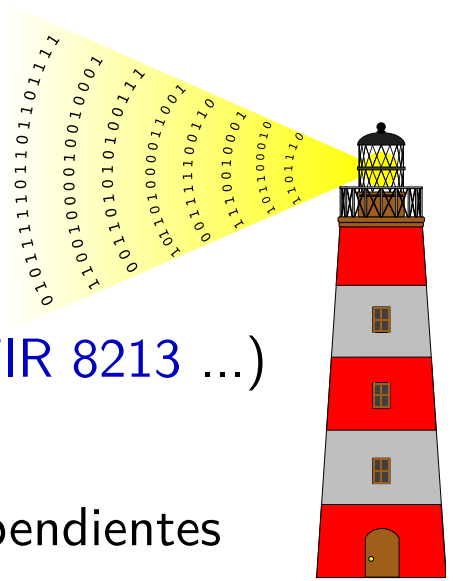
https://csrc.nist.gov/Projects/Interoperable-Randomness-Beacons

Faros de aleatoriedad interoperables

Proyecto NIST

Un Faro de aliatoriedad — a grandes rasgos:

- Un servicio que produce aleatoriedad pública en forma periódica
- *Pulso* periódico de aleatoriedad
- Cada pulso contiene una secuencia de 512 bits aleatorios recién generados
- Los pulsos están indexados, tienen un sello de tiempo y una firma digital
- Pulsos pasados son de acceso público
- La secuencia de pulsos forma una cadena de hash



El proyecto:

A. promover una referencia para faros interoperables (NISTIR 8213 ...)
B. mantener una implementación de faro en NIST
C. promover el despliegue de faros por organizaciones independientes
D. promover el uso de la aleatoriedad de faros (auditabilidad pública, ...)

https://csrc.nist.gov/Projects/Interoperable-Randomness-Beacons

Otros procesos (ejemplos)

El grupo de crypto tiene otros proyectos en curso, por ejemplo:

Otros procesos (ejemplos)

El grupo de crypto tiene otros proyectos en curso, por ejemplo:

- Lightweight cryptography (ciphers, auth-encryption, hash functions)
- Circuit Complexity (multiplicative complexity, ...)
- Digital Signatures
- Random Bit Generation
- ... https://www.nist.gov/itl/csd/cryptographic-technology

Otros procesos (ejemplos)

El grupo de crypto tiene otros proyectos en curso, por ejemplo:

- Lightweight cryptography (ciphers, auth-encryption, hash functions)
- Circuit Complexity (multiplicative complexity, ...)
- Digital Signatures
- Random Bit Generation
- ... https://www.nist.gov/itl/csd/cryptographic-technology

El resto de la charla toca brevemente en:

- Privacy-Enhancing Cryptography
- Post-quantum Cryptography

Otros procesos (ejemplos)

El grupo de crypto tiene otros proyectos en curso, por ejemplo:

- Lightweight cryptography (ciphers, auth-encryption, hash functions)
- Circuit Complexity (multiplicative complexity, ...)
- Digital Signatures
- Random Bit Generation
- ... https://www.nist.gov/itl/csd/cryptographic-technology

El resto de la charla toca brevemente en:

- Privacy-Enhancing Cryptography
- Post-quantum Cryptography

Proceso de desarrollo:

- NISTIR 7977: NIST Cryptographic Standards and Guidelines Development Process (2016). Varios principios formalizados: transparencia, apertura, equilibrio, integridad, mérito técnico, usabilidad, aceptabilidad global, mejora continua, innovación y propiedad intelectual (and consideraciones generales)

Pauta 2

Criptografía de mejora de la privacidad (PEC)

Proyecto NIST

Seguir el progreso de las tecnologías emergentes en PEC

- Ha estado inactivo ... ahora revivido.
- Rol fundamental de SMPC (**c**ómputo **s**eguro **m**ulti**p**artita) y ZKPs (**p**ruebas de nula divulgación (***z**ero-**k**nowledge **p**roof*).
- Un objetivo importante: desarrollar **materiales de referencia**

https://csrc.nist.gov/Projects/Privacy-Enhancing-Cryptography

Criptografía de mejora de la privacidad (PEC)

Proyecto NIST

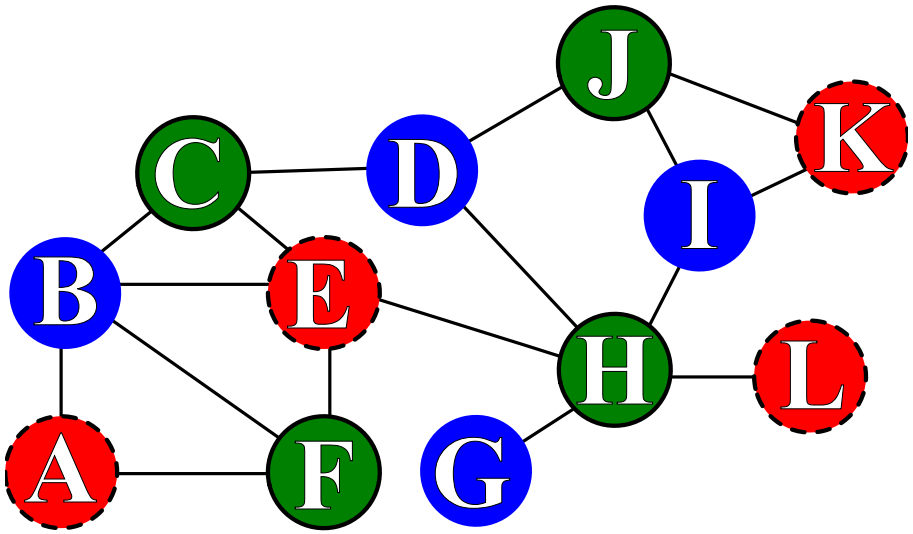
Seguir el progreso de las tecnologías emergentes en PEC

- Ha estado inactivo ... ahora revivido.
- Rol fundamental de SMPC (**c**ómputo **s**eguro **m**ulti**p**artita) y ZKPs (**p**ruebas de nula divulgación (***z**ero-**k**nowledge **p**roof*).
- Un objetivo importante: desarrollar **materiales de referencia**
- Algunos usos de interés:
  - Identificación intermediada
  - Auditabilidad pública que preserva la privacidad
  - "Derecho de los estudiantes a saber"
  - ...

https://csrc.nist.gov/Projects/Privacy-Enhancing-Cryptography

Pruebas de nula divulgación (ZKPs)

Ejemplo [GMW'91]: cómo demostrar el conocimiento de una tricolor de gráficos, sin revelar ninguna información?

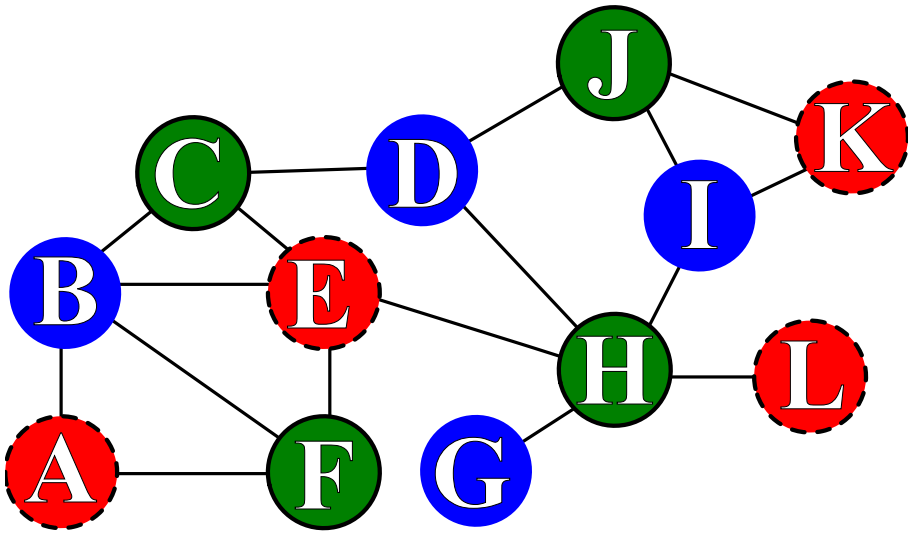


Ejemplo:

- 12 vértices: {A,B,C, D,E,F,G,H,I,J,K,L}
- 17 arcos: {AB,AF,BC, BE,BF,CD,CE,DH,DJ,EF, EH,GH,HI,HL,IJ,IK,JK}

Pruebas de nula divulgación (ZKPs)

Ejemplo [GMW'91]: cómo demostrar el conocimiento de una tricolor de gráficos, sin revelar ninguna información?



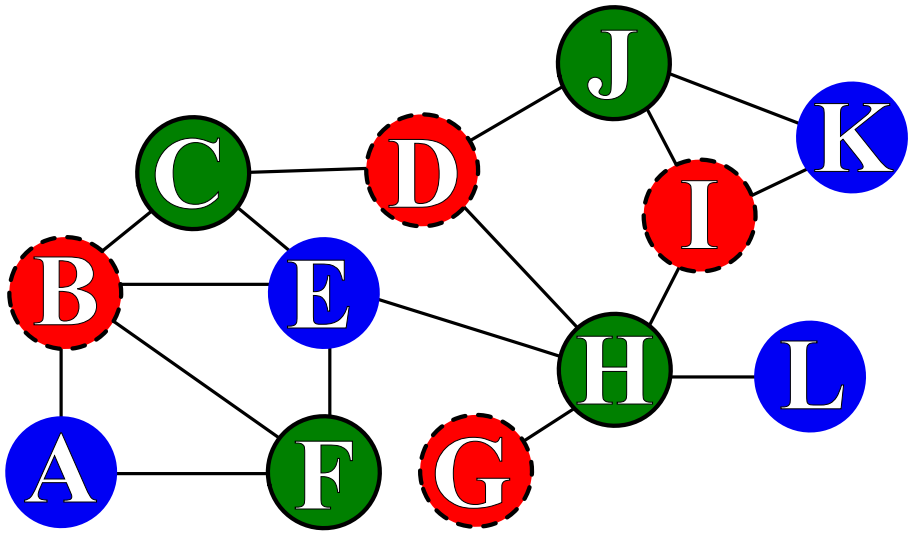
Ejemplo:

- 12 vértices: {A,B,C, D,E,F,G,H,I,J,K,L}
- 17 arcos: {AB,AF,BC, BE,BF,CD,CE,DH,DJ,EF, EH,GH,HI,HL,IJ,IK,JK}

Muchas iteraciones de lo siguiente:

Pruebas de nula divulgación (ZKPs)

Ejemplo [GMW'91]: cómo demostrar el conocimiento de una tricolor de gráficos, sin revelar ninguna información?



Ejemplo:

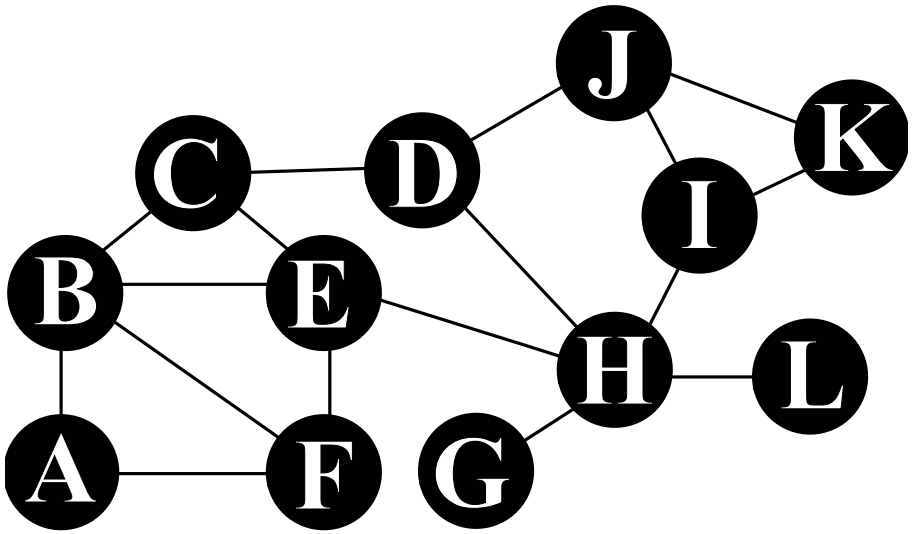
- 12 vértices: {A,B,C, D,E,F,G,H,I,J,K,L}
- 17 arcos: {AB,AF,BC, BE,BF,CD,CE,DH,DJ,EF, EH,GH,HI,HL,IJ,IK,JK}

Muchas iteraciones de lo siguiente:

- Demostrador permuta aleatoriamente los colores (e.g., ●●●→●●●);

Pruebas de nula divulgación (ZKPs)

Ejemplo [GMW'91]: cómo demostrar el conocimiento de una tricolor de gráficos, sin revelar ninguna información?



Ejemplo:

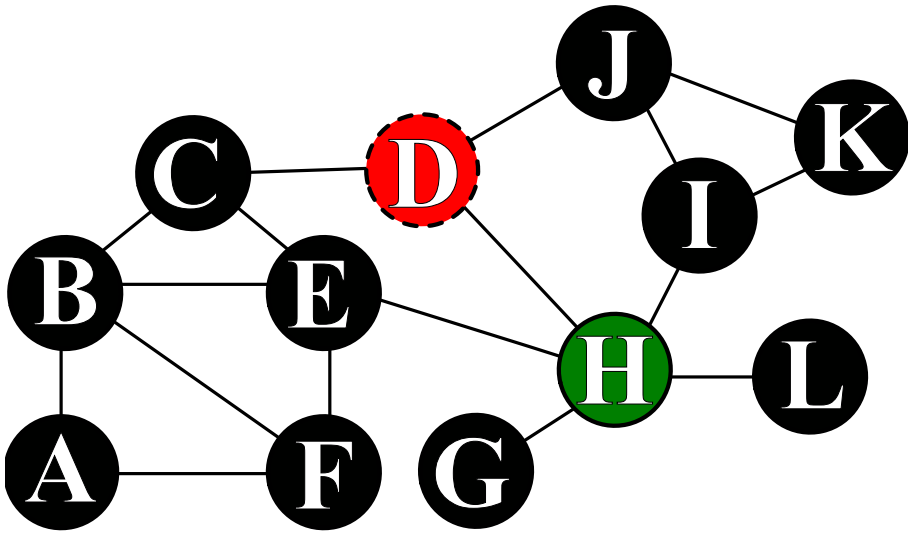
- 12 vértices: {A,B,C, D,E,F,G,H,I,J,K,L}
- 17 arcos: {AB,AF,BC, BE,BF,CD,CE,DH,DJ,EF, EH,GH,HI,HL,IJ,IK,JK}

Muchas iteraciones de lo siguiente:

- Demostrador permuta aleatoriamente los colores (e.g., ●●●→●●●);
- Demostrador se compromete a los colores de todos los vértices;

Pruebas de nula divulgación (ZKPs)

Ejemplo [GMW'91]: cómo demostrar el conocimiento de una tricolor de gráficos, sin revelar ninguna información?



Ejemplo:

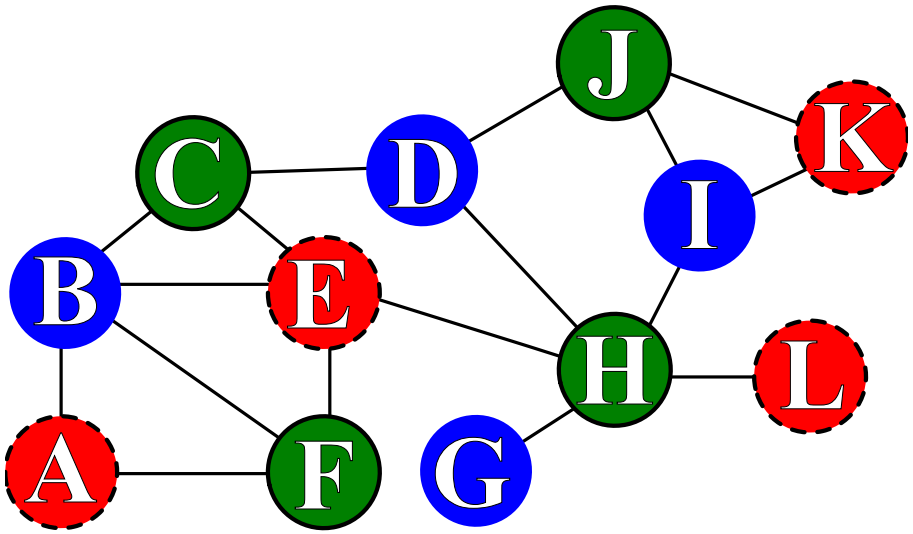
- 12 vértices: {A,B,C, D,E,F,G,H,I,J,K,L}
- 17 arcos: {AB,AF,BC, BE,BF,CD,CE,DH,DJ,EF, EH,GH,HI,HL,IJ,IK,JK}

Muchas iteraciones de lo siguiente:

- Demostrador permuta aleatoriamente los colores (e.g., ●●●→●●●);
- Demostrador se compromete a los colores de todos los vértices;
- El verificador selecciona un arco aleatorio y el demostrador abre sus dos vértices.

Pruebas de nula divulgación (ZKPs)

Ejemplo [GMW'91]: cómo demostrar el conocimiento de una tricolor de gráficos, sin revelar ninguna información?



Ejemplo:

- 12 vértices: {A,B,C, D,E,F,G,H,I,J,K,L}
- 17 arcos: {AB,AF,BC, BE,BF,CD,CE,DH,DJ,EF, EH,GH,HI,HL,IJ,IK,JK}

Muchas iteraciones de lo siguiente:

- Demostrador permuta aleatoriamente los colores (e.g., ●●●→●●●);
- Demostrador se compromete a los colores de todos los vértices;
- El verificador selecciona un arco aleatorio y el demostrador abre sus dos vértices.

El verificador acepta si y solo si los arcos tienen 2 colores diferentes

Cómputo **s**eguro **m**ulti**p**artita (SMPC)

Varias partes calculan conjuntamente una función sobre sus datos y mantienen la privacidad de esos datos

Cómputo **s**eguro **m**ulti**p**artita (SMPC)

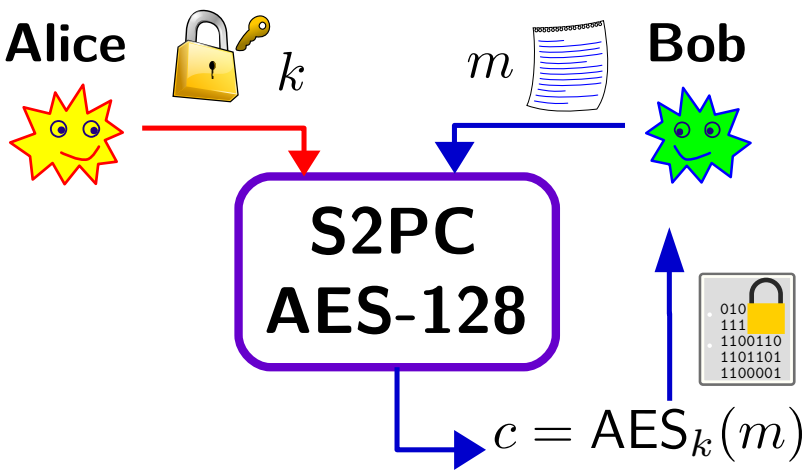
Varias partes calculan conjuntamente una función sobre sus datos y mantienen la privacidad de esos datos (la definición técnica es más complicada)

Cómputo seguro multipartita (SMPC)

Varias partes calculan conjuntamente una función sobre sus datos y mantienen la privacidad de esos datos (la definición técnica es más complicada)

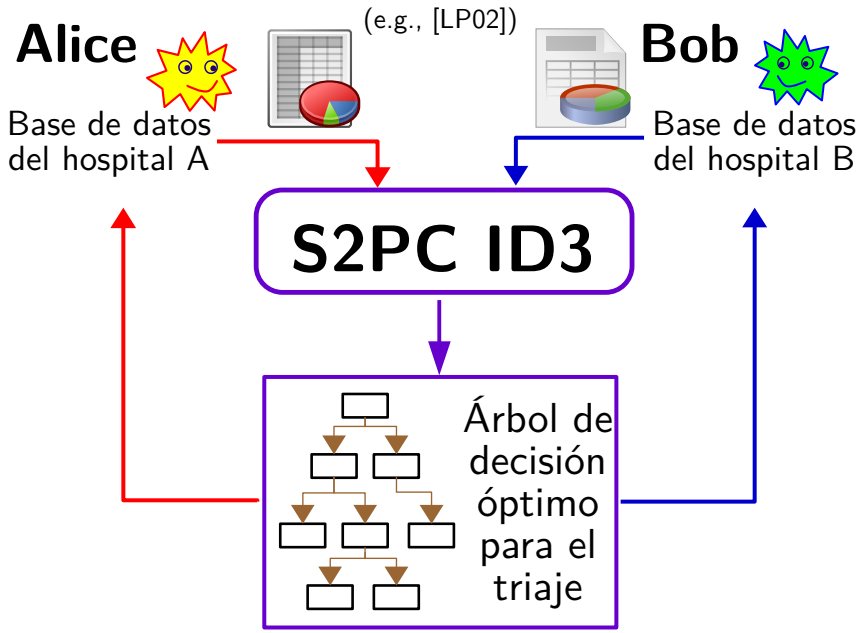
Ejemplos de S2PC:

Cifrado ciego (e.g., [PSSW09])
(Ejemplo típico de referencia)



AES-128: **e**stándar de **c**ifrado **a**vanzado (un cifrado de bloque) con 128 bits de tamaño de clave y de texto en claro.

Minería de datos con preservación de la privacidad



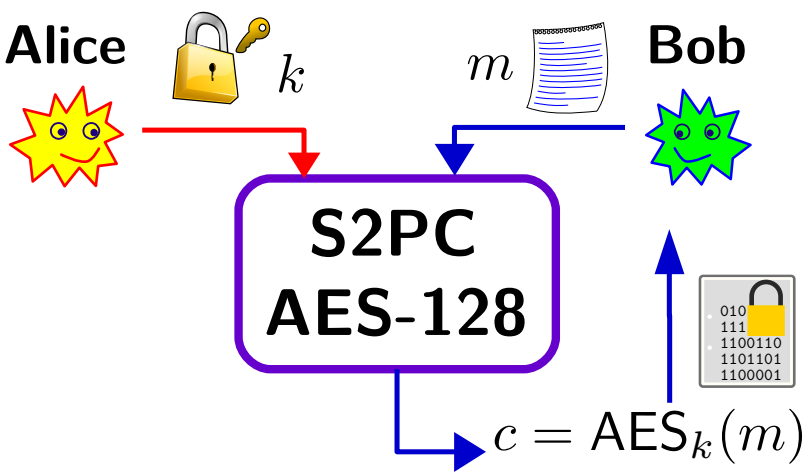
ID3: dicotomizador iterativo 3: un algoritmo para generar un árbol de decisión a partir de un conjunto de datos

Cómputo seguro multipartita (SMPC)

Varias partes calculan conjuntamente una función sobre sus datos y mantienen la privacidad de esos datos (la definición técnica es más complicada)

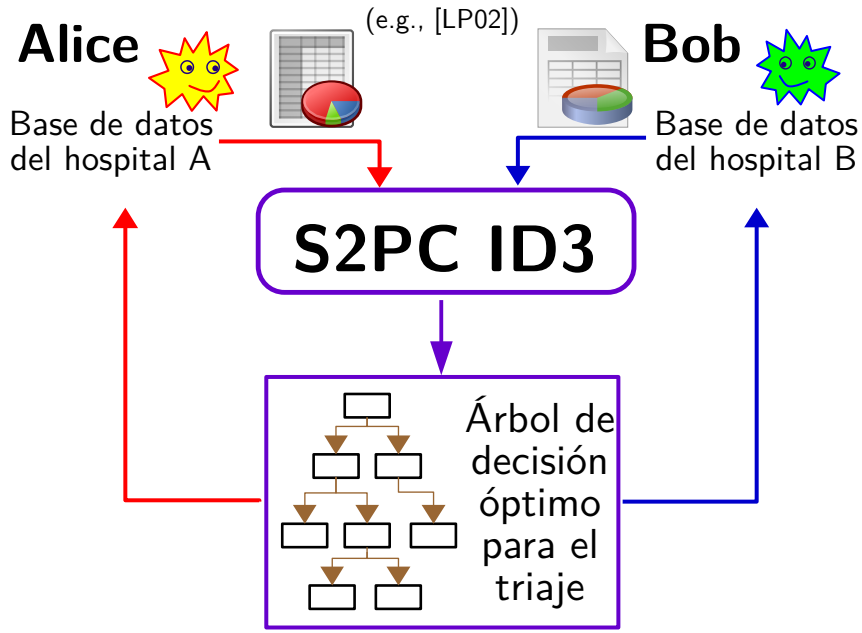
Ejemplos de S2PC:

Cifrado ciego (e.g., [PSSW09])
(Ejemplo típico de referencia)



AES-128: **e**stándar de **c**ifrado **a**vanzado (un cifrado de bloque) con 128 bits de tamaño de clave y de texto en claro.

Minería de datos con preservación de la privacidad

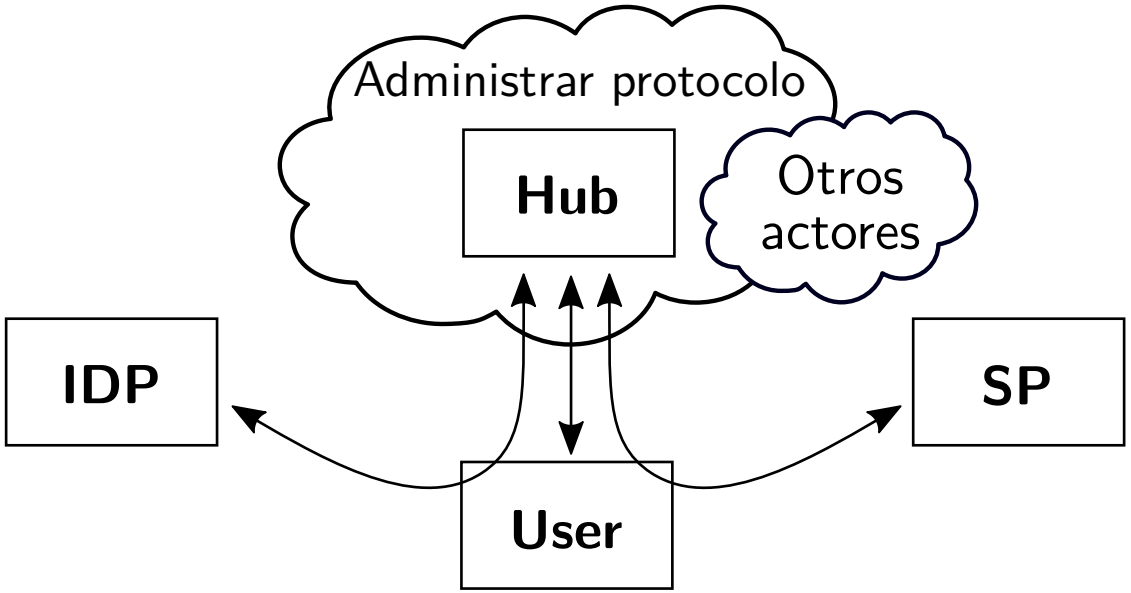


ID3: dicotomizador iterativo 3: un algoritmo para generar un árbol de decisión a partir de un conjunto de datos

Puede verse como una generalización de una prueba de nula divulgación

Identificación intermediada (1/3)

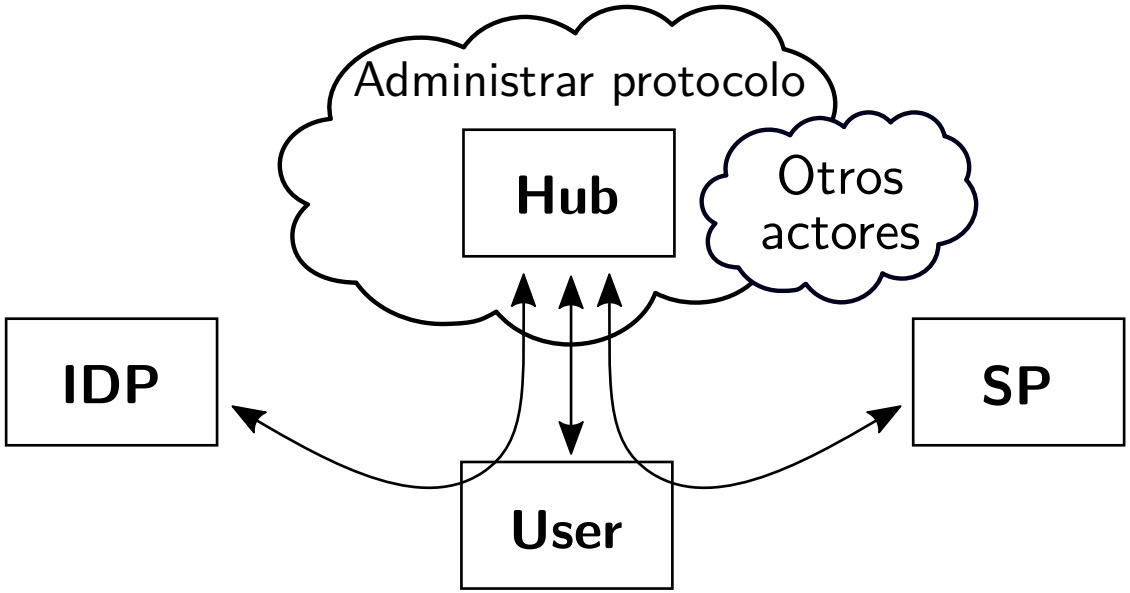
Caso de uso



Legend: IDP (**p**roveedor de **id**entidad); SP (**p**roveedor de **s**ervicio)

Identificación intermediada (1/3)

Caso de uso

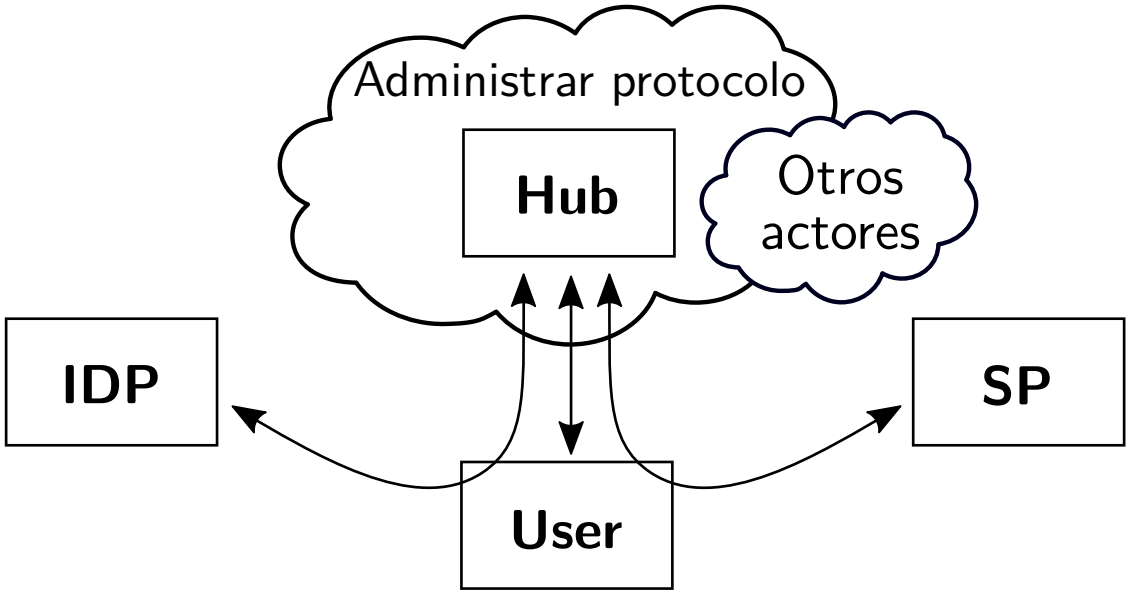


Legend: IDP (**p**roveedor de **id**entidad); SP (**p**roveedor de **s**ervicio)

- ¿Por qué este ejemplo? Se relaciona con la privacidad y con identidad.

Identificación intermediada (1/3)

Caso de uso

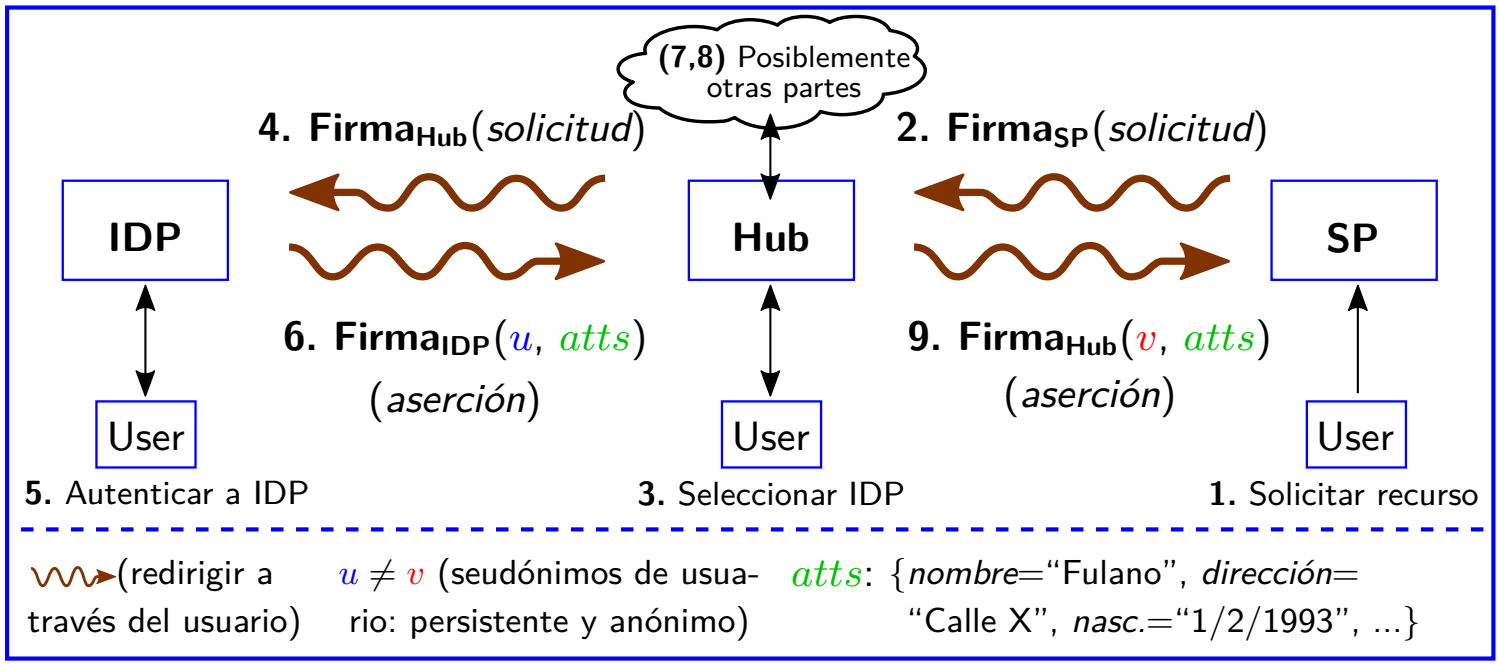


Legend: IDP (**p**roveedor de **id**entidad); SP (**p**roveedor de **s**ervicio)

- ¿Por qué este ejemplo? Se relaciona con la privacidad y con identidad.
- **Restricciones de diseño**: usuario mayormente pasivo; el mediador debe existir. (No siempre podemos elegir el paradigma de solución óptima.)

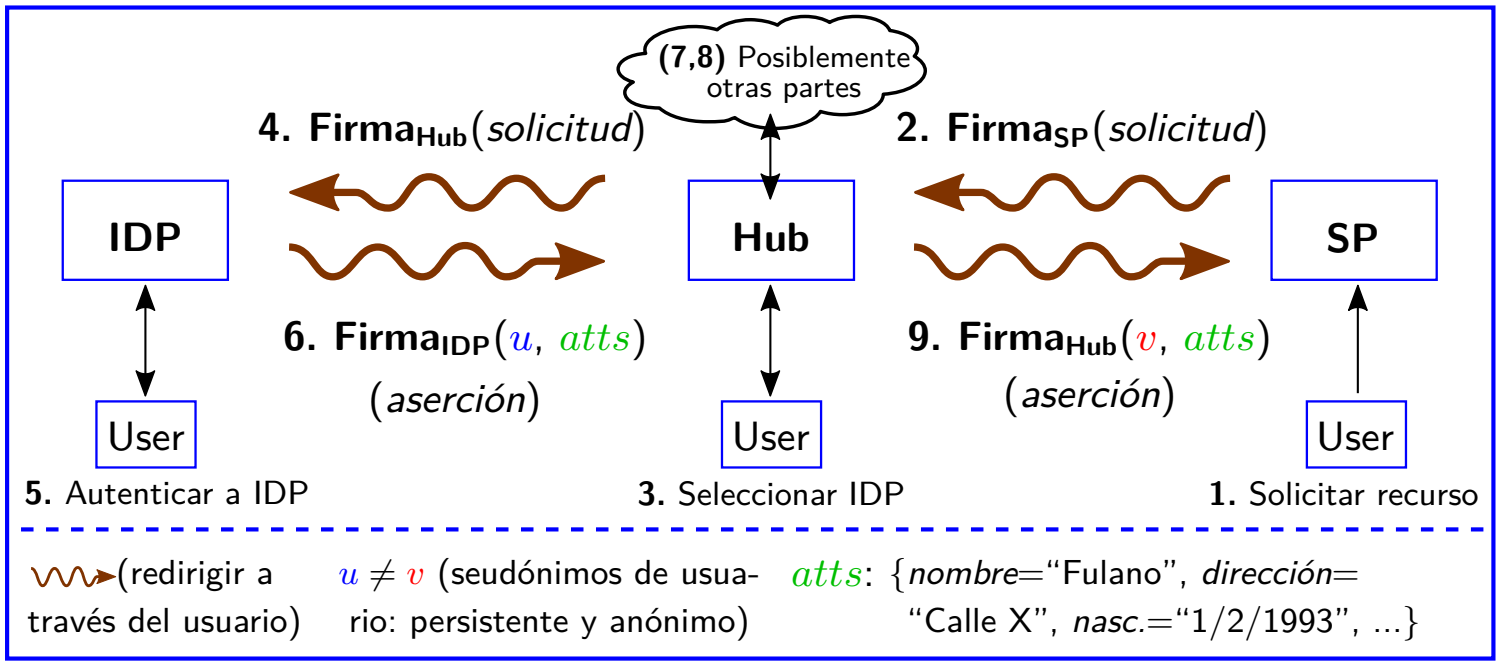
Identificación intermediada (2/3)

Caso de uso



Identificación intermediada (2/3)

Caso de uso

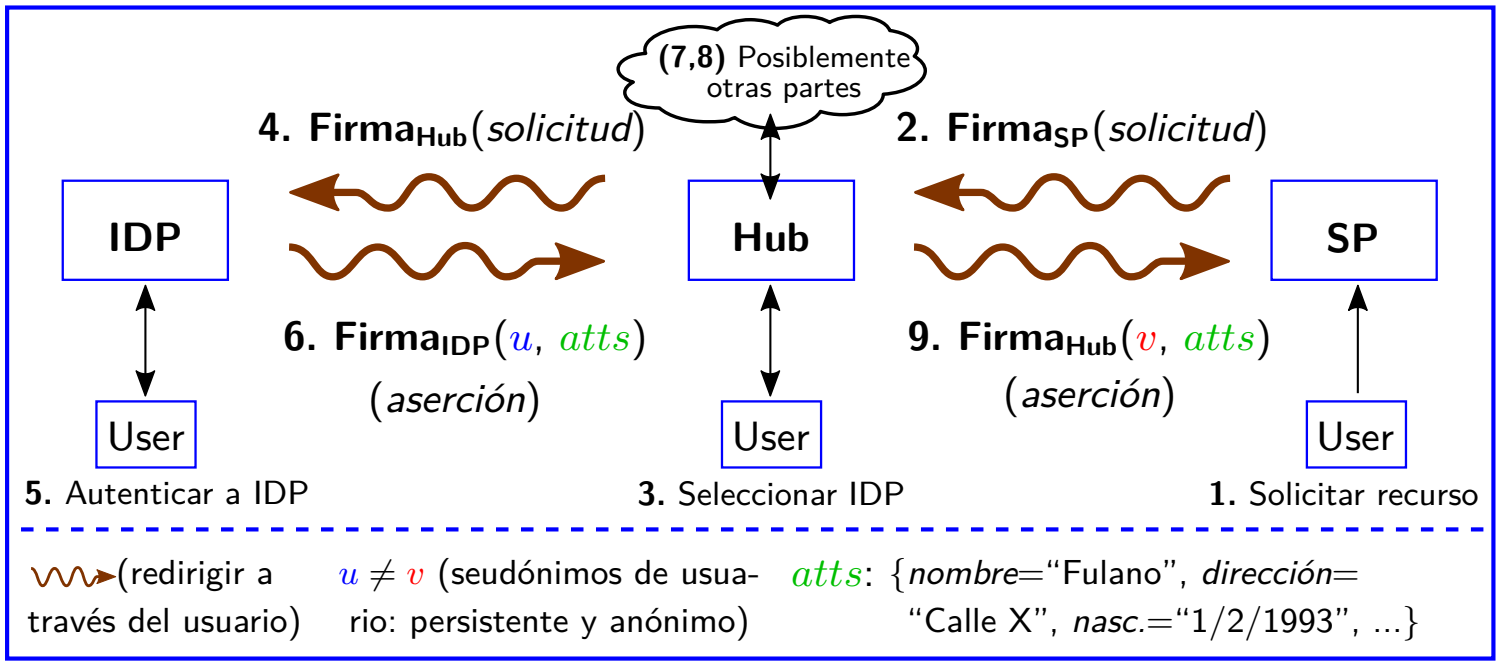


La "Estrategia nacional para identidades confiables en el ciberespacio" (NSTIC en los EE. UU.) requería propiedades de privacidad para esto, e.g.:

- Atributos cifrados de extremo a extremo
- Desvinculabilidad de transacciones del usuario por parte del Hub

Identificación intermediada (2/3)

Caso de uso



La "Estrategia nacional para identidades confiables en el ciberespacio" (NSTIC en los EE. UU.) requería propiedades de privacidad para esto, e.g.:

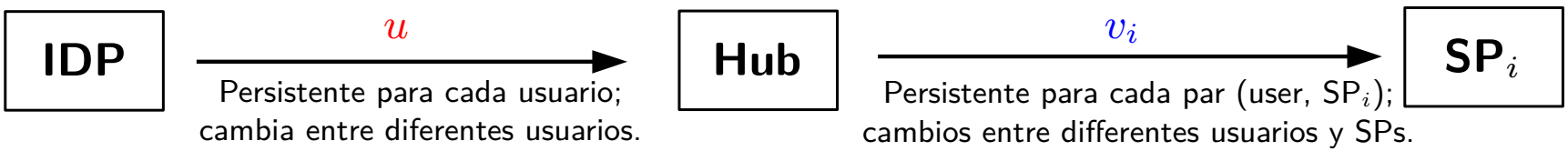
- Atributos cifrados de extremo a extremo
- Desvinculabilidad de transacciones del usuario por parte del Hub

PEC puede resolverlo ...

Identificación intermediada (3/3)

Caso de uso

Sistemas actuales: Hub transforma localmente los seudónimos de usuario ($u \rightarrow v_i$)



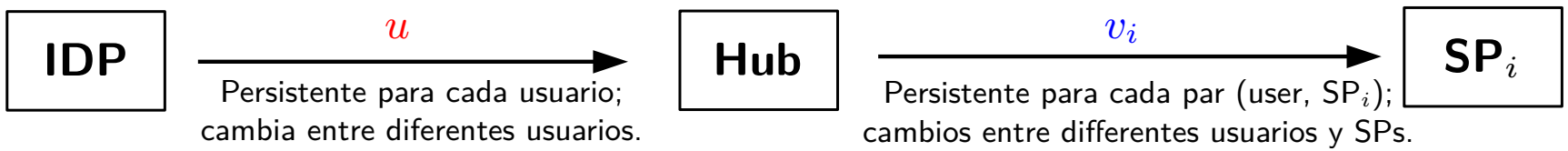
Desafío: Cómo evitar que el Hub vea u?

Se omiten algunos detalles técnicos para simplificar la presentación.

Identificación intermediada (3/3)

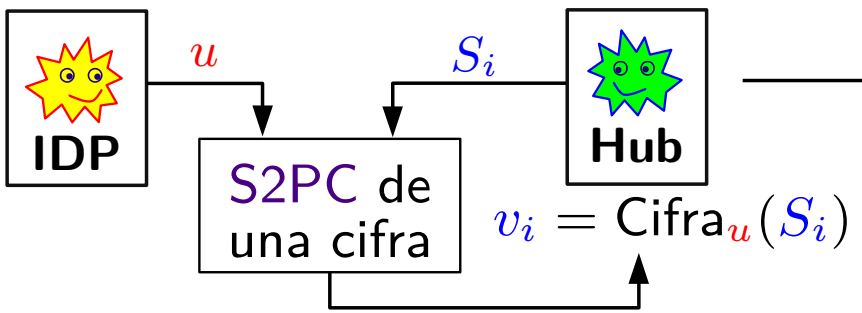
Caso de uso

Sistemas actuales: Hub transforma localmente los seudónimos de usuario ($u \to v_i$)



Desafío: Cómo evitar que el Hub vea u?

Posible mejora: Transformar los seudónimos usando un S2PC de una función de cifra



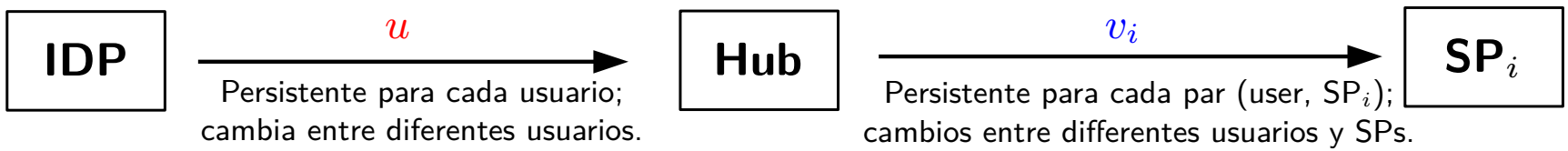
- v_i es pseudoaleatorio
- el Hub no se entera de nada sobre u
- el IDP no se entera nada sobre S_i and v_i

Se omiten algunos detalles técnicos para simplificar la presentación.

Identificación intermediada (3/3)

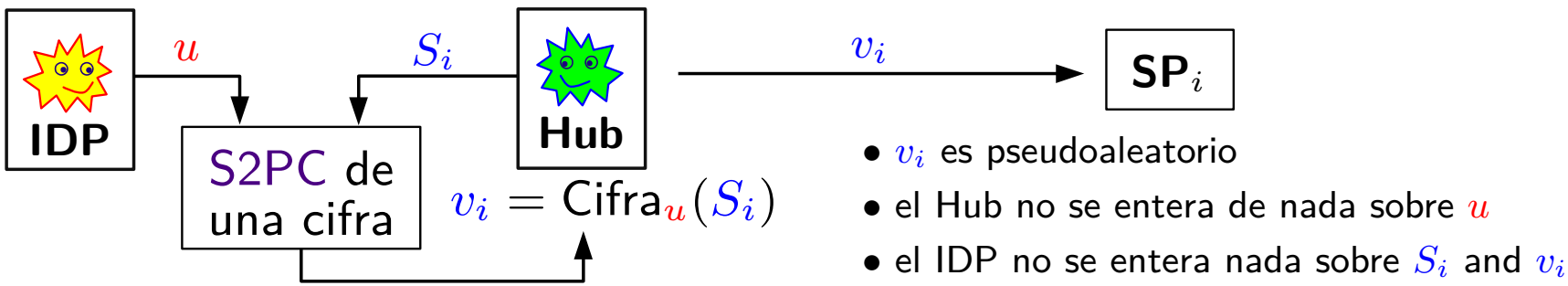
Caso de uso

Sistemas actuales: Hub transforma localmente los seudónimos de usuario ($u \rightarrow v_i$)



Desafío: Cómo evitar que el Hub vea u?

Posible mejora: Transformar los seudónimos usando un S2PC de una función de cifra



- v_i es pseudoaleatorio
- el Hub no se entera de nada sobre u
- el IDP no se entera nada sobre S_i and v_i

Cuando el mismo usuario accede a diferentes SP, el Hub ve diferentes seudónimos, **evitando así la vinculación** de transacciones del mismo usuario a través de diferentes SP.

Se omiten algunos detalles técnicos para simplificar la presentación.

Auditabilidad pública usando aleatoriedad pública (1/2)

Caso de uso

La aleatoriedad pública facilita la auditabilidad pública de los procesos aleatorios.

Mejorarlo con *privacidad* es una cuestión de PEC

Una aplicación en Chile:

Auditabilidad pública usando aleatoriedad pública (1/2)

Caso de uso

La aleatoriedad pública facilita la auditabilidad pública de los procesos aleatorios.

Mejorarlo con *privacidad* es una cuestión de PEC

Una aplicación en Chile:

- La Contraloría General de la República selecciona, al azar, a funcionarios públicos para auditorías financieras.
- Es natural que una persona seleccionada cuestione como fué la selección.
- Junto con la Univ. de Chile se ha desarrollado una aplicación que permitirá hacer la selección utilizando aleatoriedad pública.
- En etapa de prueba en Contraloría https://random.uchile.cl/projects/contraloria/

Auditabilidad pública usando aleatoriedad pública (2/2)

Caso de uso

Desafío: selección aleatoria dependiente de atributos privados

Público		Privado inherente			Privado derivado	
# (i)	Rand id	Name (N)	a_1	a_2	Weight (w)	Acc. (W)
1	371	**C**ira	1	2	0.1	0.1
2	942	**E**ron	2	7	0.3	0.4
3	107	**B**eto	1	5	0.2	0.6
4	527	**A**ldo	1	9	0.3	0.9
5	123	**D**ago	3	1	0.1	1.0

Auditabilidad pública usando aleatoriedad pública (2/2)

Caso de uso

Desafío: selección aleatoria dependiente de atributos privados

Público		Privado inherente			Privado derivado	
# (i)	Rand id	Name (N)	a_1	a_2	Weight (w)	Acc. (W)
1	371	**C**ira	1	2	0.1	0.1
2	942	**E**ron	2	7	0.3	0.4
3	107	**B**eto	1	5	0.2	0.6
4	527	**A**ldo	1	9	0.3	0.9
5	123	**D**ago	3	1	0.1	1.0

Comprométete con todos los atributos y muestra la tabla

Auditabilidad pública usando aleatoriedad pública (2/2)

Caso de uso

Desafío: selección aleatoria dependiente de atributos privados

Público		Privado inherente			Privado derivado	
# (i)	Rand id	Name (N)	a_1	a_2	Weight (w)	Acc. (W)
1	371	**C**ira	1	2	0.1	0.1
2	942	**E**ron	2	7	0.3	0.4
3	107	**B**eto	1	5	0.2	0.6
4	527	**A**ldo	1	9	0.3	0.9
5	123	**D**ago	3	1	0.1	1.0

Comprométete con todos los atributos y muestra la tabla ... después **prueba en ZK**:

1. $a_i \in A$ (e.g., nivel salarial); $b_i \in B$ (e.g., años en ese puesto);
2. $w_i = f(a_i, b_i)$ (cálculo correcto del peso);
3. $\sum_i w_i = 1$ (suma correcta de probabilidades);
4. $W_i = w_i + W_{i-1}$ (acumulador correcto de probabilidades);
5. $\{N_i\}$ =NOMBRES (nombres no repetidos de un conjunto apropiado); ...

Auditabilidad pública usando aleatoriedad pública (2/2)

Caso de uso

Desafío: selección aleatoria dependiente de atributos privados

Público		Privado inherente			Privado derivado	
# (i)	Rand id	Name (N)	a_1	a_2	Weight (w)	Acc. (W)
1	371	**C**ira	1	2	0.1	0.1
2	942	**E**ron	2	7	0.3	0.4
3	107	**B**eto	1	5	0.2	0.6
4	527	**A**ldo	1	9	0.3	0.9
5	123	**D**ago	3	1	0.1	1.0

Comprométete con todos los atributos y muestra la tabla ... después **prueba en ZK**:

1. $a_i \in A$ (e.g., nivel salarial); $b_i \in B$ (e.g., años en ese puesto);
2. $w_i = f(a_i, b_i)$ (cálculo correcto del peso);
3. $\sum_i w_i = 1$ (suma correcta de probabilidades);
4. $W_i = w_i + W_{i-1}$ (acumulador correcto de probabilidades);
5. $\{N_i\}$ =NOMBRES (nombres no repetidos de un conjunto apropiado); ...

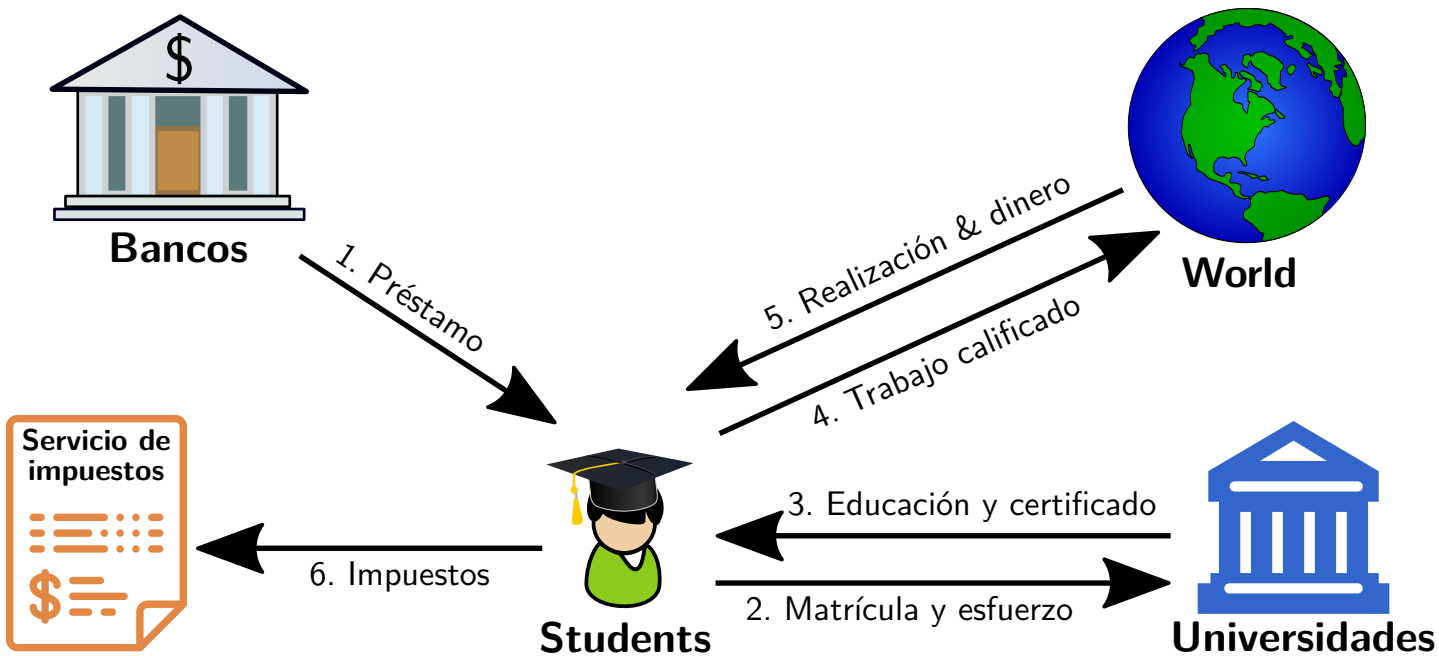
Deriva $R : 0 < R \leq 1$ (aleatorio) del faro y determina # j: $W_{\text{máx}(1,j-1)} < R \leq W_j$

- **Prueba en ZK** que j es consistente con R y la tabla de compromisos

Derecho del alumno a saber

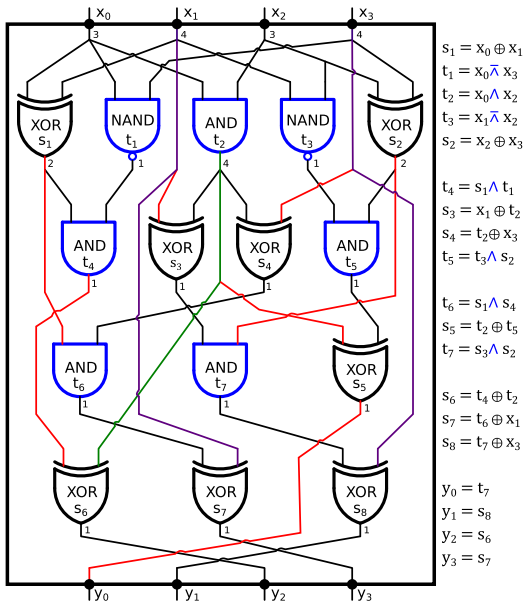
Caso de uso

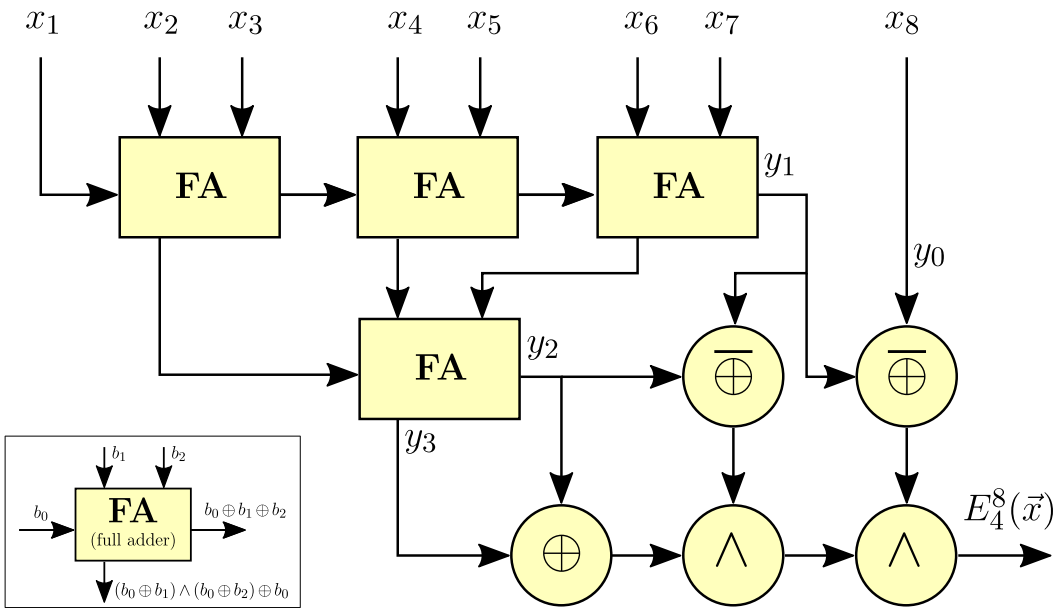
- Propuesta para ordenar el uso de SMPC para calcular el retorno monetario de la inversión del estudiante en educación.
- Los datos se distribuyen entre varias entidades. Debido a preocupaciones de privacidad, estas entidades no pueden compartir los datos.
- https://www.govtrack.us/congress/bills/116/s681/text



Investigación en **c**omplejidad **m**ultiplicativa (MC)

- Circuitos de referencia para AES
- MC es relevante para ZK, SMPC, ..., ya que generalmente las compuertas XOR son gratuitas y las compuertas AND son caras
- Intención de desarrollar un formato de archivo de circuito





Pauta 3

Computadoras cuánticas

- Mucho más poderosas que las computadoras clásicas.

Computadoras cuánticas

- Mucho más poderosas que las computadoras clásicas.
- Probablemente no podrán resolver problemas NP-difíciles.

Computadoras cuánticas

- Mucho más poderosas que las computadoras clásicas.
- Probablemente no podrán resolver problemas NP-difíciles.
- Pueden factorizar enteros, resolver logaritmo discreto (incluyendo en curvas elípticas).

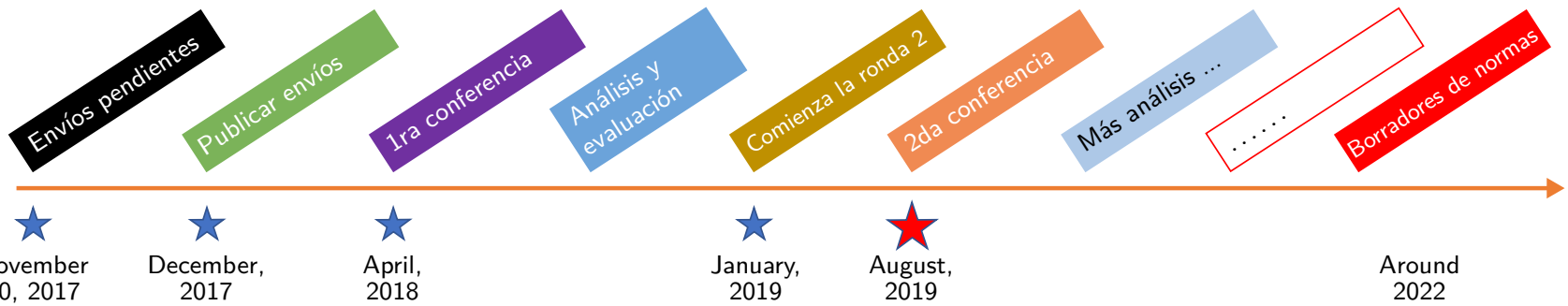
Computadoras cuánticas

- Mucho más poderosas que las computadoras clásicas.
- Probablemente no podrán resolver problemas NP-difíciles.
- Pueden factorizar enteros, resolver logaritmo discreto (incluyendo en curvas elípticas).
- Búsqueda en tiempo $O(\sqrt{n})$ en un espacio de tamaño n.

Criptografía post-cuántica (PQC)

Proyecto NIST

- Alcance: firmas digitales, cifrado de clave pública y KEM
- La ronda 1 finalizó el 30 de enero de 2019
- 26 propuestas en la Ronda 2
  - 17 Cifrado/KEMs (9 retículos, 7 códigos, 1 isogenia)
  - 9 firmas (4 multivariadas, 3 retículos, 2 simétricas)
  - Las especificaciones / códigos para las propuestas están disponibles en www.nist.gov/pqcrypto



https://csrc.nist.gov/projects/post-quantum-cryptography

Traducido de la transparencia, preparada por Dustin Moody, con actualización sobre el proyecto PQC

Situación actual

Estimación: En 15 años, usando una computadora cuántica podremos factorizar un número RSA de 2Kb en aproximadamente un día, usando una planta de energía nuclear dedicada. (Mariantoni, PQCrypto 2014)

Situación actual

Estimación: En 15 años, usando una computadora cuántica podremos factorizar un número RSA de 2Kb en aproximadamente un día, usando una planta de energía nuclear dedicada. (Mariantoni, PQCrypto 2014)

Si es razonable preocuparse por esto: entonces es urgente encontrar alternativas a los sistemas actuales de clave pública: firmas, cifrado y acuerdo de clave.

- Históricamente, la adopción de criptografía nueva ha tomado del orden de una década o más.
- Comunicaciones cifradas actuales podrían estar siendo grabadas para ser descifradas una vez que se cuente con computadores cuánticos.

Situación actual

Estimación: En 15 años, usando una computadora cuántica podremos factorizar un número RSA de 2Kb en aproximadamente un día, usando una planta de energía nuclear dedicada. (Mariantoni, PQCrypto 2014)

Si es razonable preocuparse por esto: entonces es urgente encontrar alternativas a los sistemas actuales de clave pública: firmas, cifrado y acuerdo de clave.

- Históricamente, la adopción de criptografía nueva ha tomado del orden de una década o más.
- Comunicaciones cifradas actuales podrían estar siendo grabadas para ser descifradas una vez que se cuente con computadores cuánticos.

Hay varias alternativas propuestas:

https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions

- Basadas en códigos, hash, isogenia, retículos, multivariada, ...
- ¿Qué tan bien entendemos estas herramientas?
- ¿Cual es el riesgo de estandarizar estas técnicas alternativas?

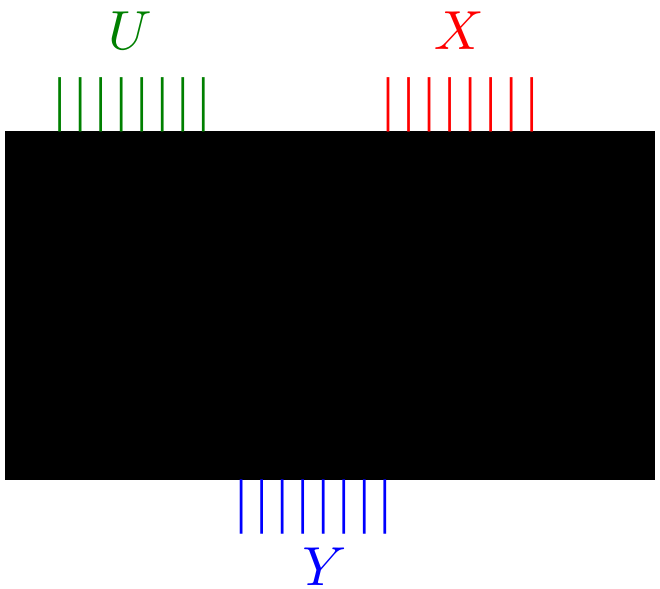
La propuesta "Picnic"

https://microsoft.github.io/Picnic/

Puntos claves:

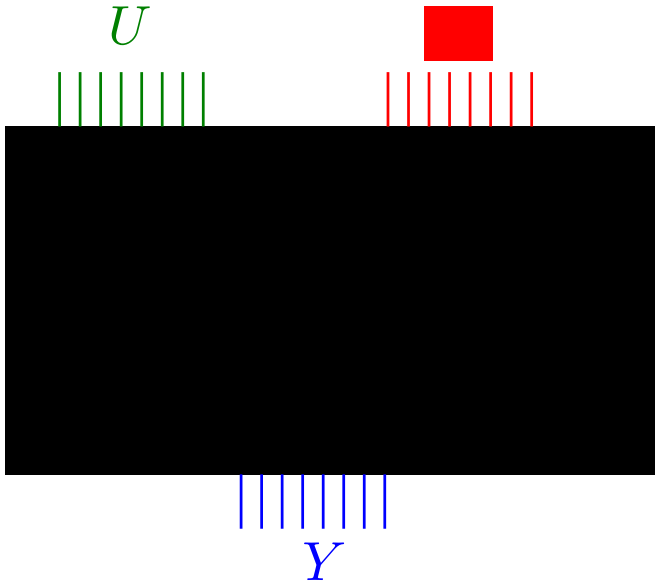
- Un esquema de firma (generar clave; firmar; verificar)
- Sin suposiciones de dureza teórica o estructurada
- Reducción de seguridad a primitivas simétricas:
  - Hash (SHA3)
  - Cifra de bloque (Low MC ... también podría ser AES)
  - PRNG (SHAKE)
- Construcción basada en una ZKPoK[1], usando:
  - Protocolo Σ (sigma)
  - Transformaciones para la no interactividad: Fiat-Shamir y Unruh
  - "MPC en la cabeza"
- El largo de la firma es proporcional al número de compuertas `AND` de la cifra de bloque

Picnic — ilustración a alto nivel



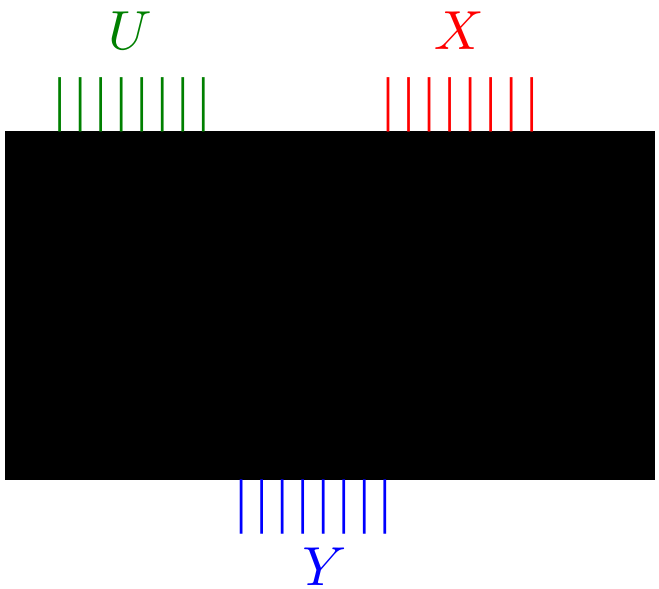
- ■ es una función unidireccional
- Dado U y Y, yo afirmo que conozco X

Picnic — ilustración a alto nivel



- ■ es una función unidireccional
- Dado U y Y, yo afirmo que conozco ■

Picnic — ilustración a alto nivel



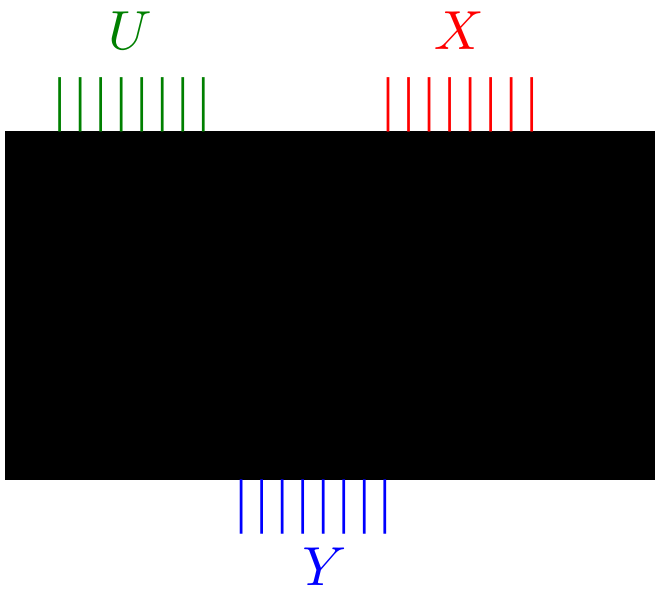
– ■ es una función unidireccional

– Dado U y Y, yo afirmo que conozco X

En Picnic:

- ■ es una función de cifrado llamada Low MC
- Y es el cifrado del texto U usando la clave X
- Esquema de firma PQC: la clave pública es (U,Y); la clave privada es X

Picnic — ilustración a alto nivel



– ■ es una función unidireccional

– Dado U y Y, yo afirmo que conozco X

En Picnic:

- ■ es una función de cifrado llamada Low MC
- Y es el cifrado del texto U usando la clave X
- Esquema de firma PQC: la clave pública es (U,Y); la clave privada es X

Necesitaremos una ZKP de conocimiento de X.

Un enfoque basado en circuitos

En un circuito para "Low MC":

1. **Para cada cable de entrada:** divide (*secret-share*) su valor booleano v en tres partes (*shares*) aleatorias a_1, a_2, a_3 tal que $v = a_1 + a_2 + a_3$

Un enfoque basado en circuitos

En un circuito para "Low MC":

1. **Para cada cable de entrada:** divide (*secret-share*) su valor booleano v en tres partes (*shares*) aleatorias a_1, a_2, a_3 tal que $v = a_1 + a_2 + a_3$

2. Envía parte a_i al jugador P_i

Un enfoque basado en circuitos

En un circuito para "Low MC":

1. **Para cada cable de entrada:** divide (*secret-share*) su valor booleano v en tres partes (*shares*) aleatorias a_1, a_2, a_3 tal que $v = a_1 + a_2 + a_3$

2. Envía parte a_i al jugador P_i

3. Para cada compuerta: propaga las *shares* desle las entradas a las salidas

Un enfoque basado en circuitos

En un circuito para "Low MC":

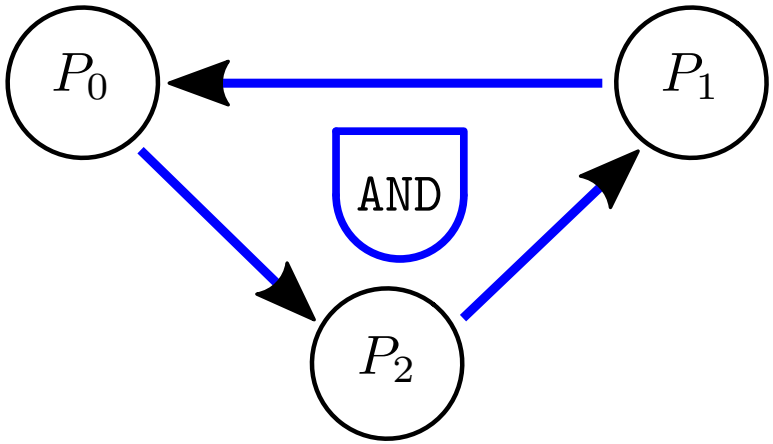
1. **Para cada cable de entrada:** divide (*secret-share*) su valor booleano v en tres partes (*shares*) aleatorias a_1, a_2, a_3 tal que $v = a_1 + a_2 + a_3$
2. Envía parte a_i al jugador P_i
3. Para cada compuerta: propaga las *shares* desle las entradas a las salidas
4. Para cada cable de salida: revela las tres *shares*

Un enfoque basado en circuitos

En un circuito para "Low MC":

1. **Para cada cable de entrada:** divide (*secret-share*) su valor booleano v en tres partes (*shares*) aleatorias a_1, a_2, a_3 tal que $v = a_1 + a_2 + a_3$
2. Envía parte a_i al jugador P_i
3. Para cada compuerta: propaga las *shares* desle las entradas a las salidas [próxima transparencia]
4. Para cada cable de salida: revela las tres *shares*

S3PC en la cabeza



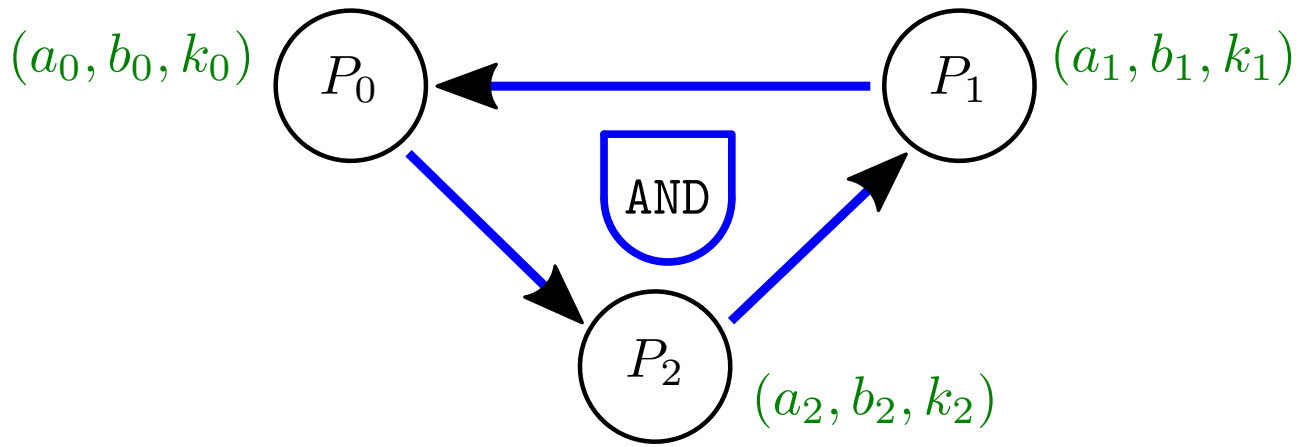
Dados:

$A = a_0 + a_1 + a_2$
$B = b_0 + b_1 + b_2$
$C = c_0 + c_1 + c_2$

Verifiquesé:

$C = A \texttt{ AND } B$

S3PC en la cabeza



Dados:

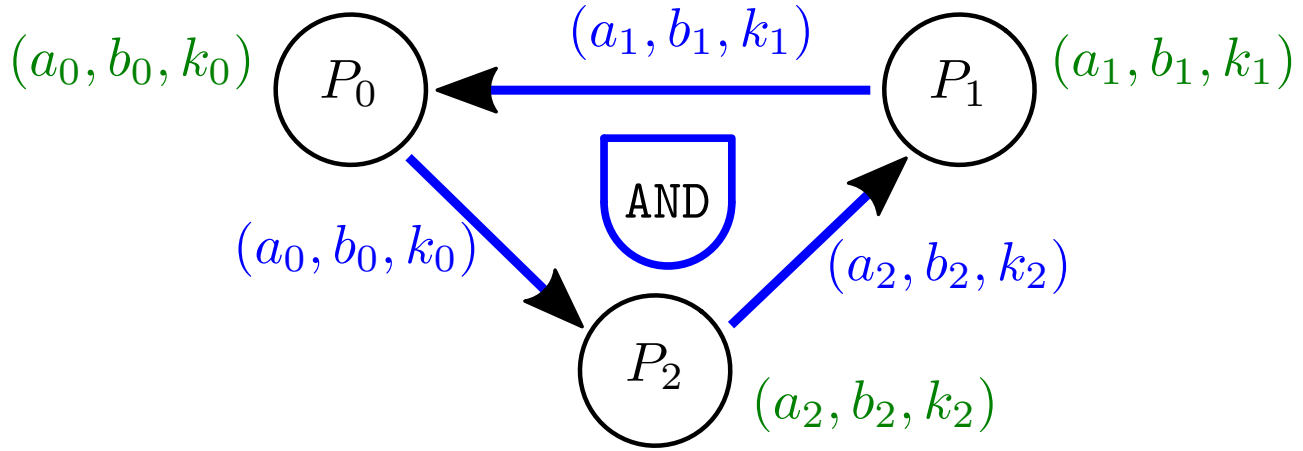
$A = a_0 + a_1 + a_2$
$B = b_0 + b_1 + b_2$
$C = c_0 + c_1 + c_2$

Verifiquesé:

$C = A \texttt{ AND } B$

1. *Shares* de entrada iniciales y bits aleatorios: a_i, b_i, k_i

S3PC en la cabeza



Dados:

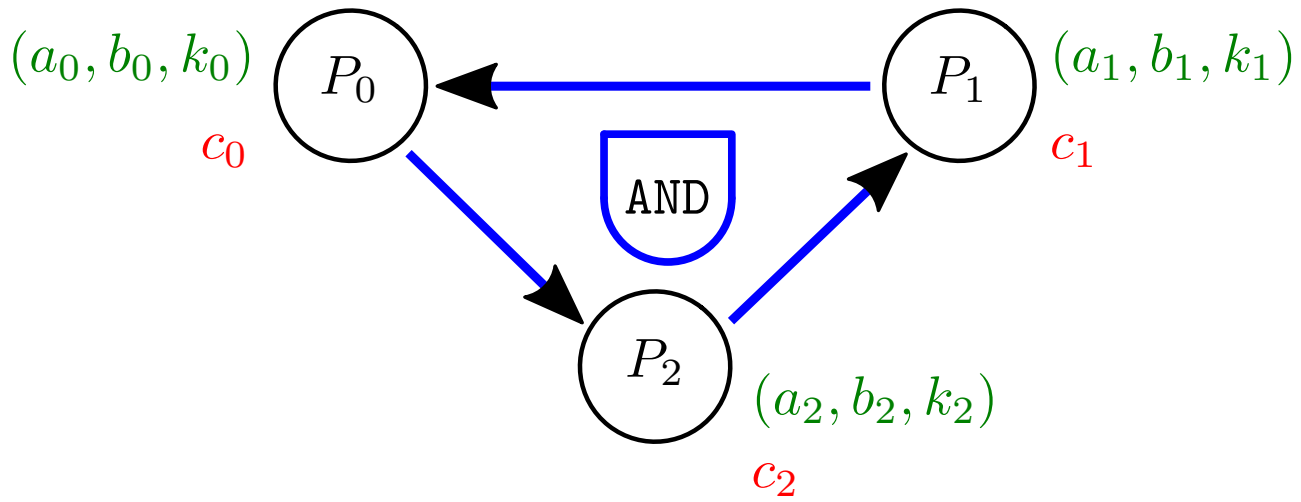
$A = a_0 + a_1 + a_2$
$B = b_0 + b_1 + b_2$
$C = c_0 + c_1 + c_2$

Verifiquesé:

$C = A \texttt{ AND } B$

1. *Shares* de entrada iniciales y bits aleatorios: a_i, b_i, k_i
2. Comunicar: $P_{i-1 \pmod 3}$ recibe (a_i, b_i, k_i)

S3PC en la cabeza



Dados:

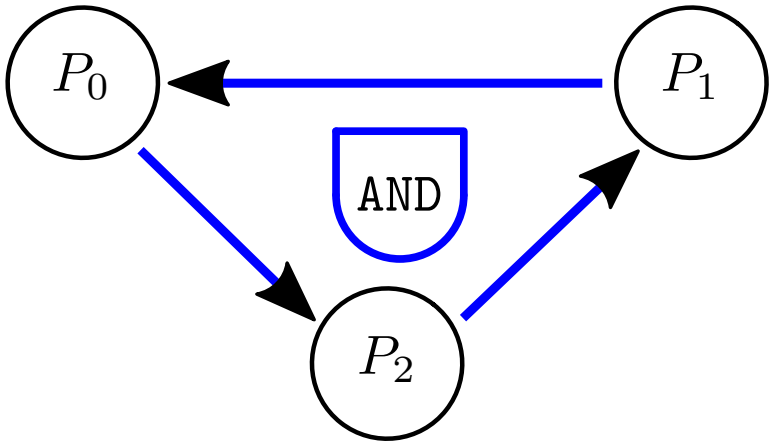
$$A = a_0 + a_1 + a_2$$
$$B = b_0 + b_1 + b_2$$
$$C = c_0 + c_1 + c_2$$

Verifiquesé:

$$C = A \texttt{ AND } B$$

1. *Shares* de entrada iniciales y bits aleatorios: a_i, b_i, k_i
2. Comunicar: $P_{i-1 \pmod 3}$ recibe (a_i, b_i, k_i)
3. Calcular *shares* de salida: $c_i = a_i b_i + a_i b_{i+1} + a_{i+1} b_i + k_i + k_{i+1}$

S3PC en la cabeza



Dados:

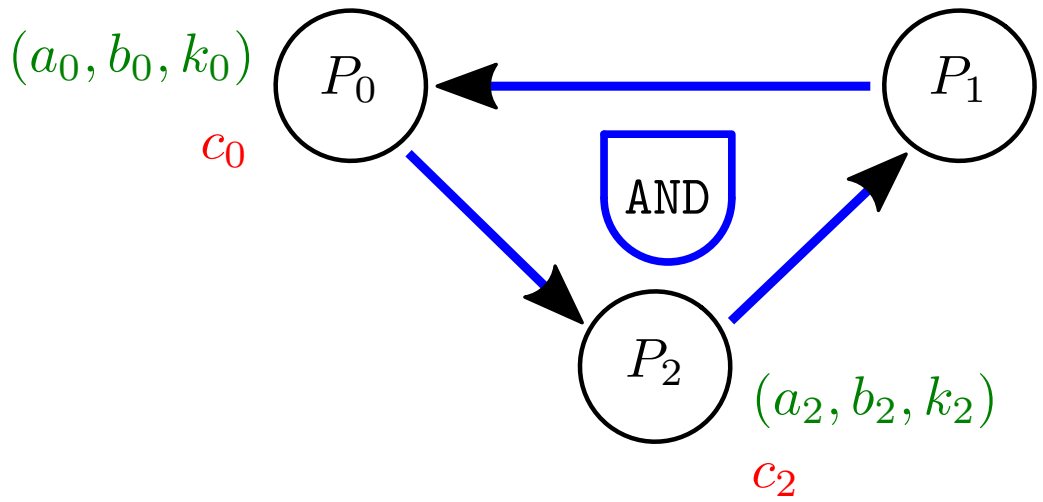
$A = a_0 + a_1 + a_2$
$B = b_0 + b_1 + b_2$
$C = c_0 + c_1 + c_2$

Verifiquesé:

$C = A \texttt{ AND } B$

1. *Shares* de entrada iniciales y bits aleatorios: a_i, b_i, k_i
2. Comunicar: $P_{i-1 (\text{mod } 3)}$ recibe (a_i, b_i, k_i)
3. Calcular *shares* de salida: $c_i = a_i b_i + a_i b_{i+1} + a_{i+1} b_i + k_i + k_{i+1}$
4. Comprometerse ■ con todos los valores

S3PC en la cabeza



Dados:

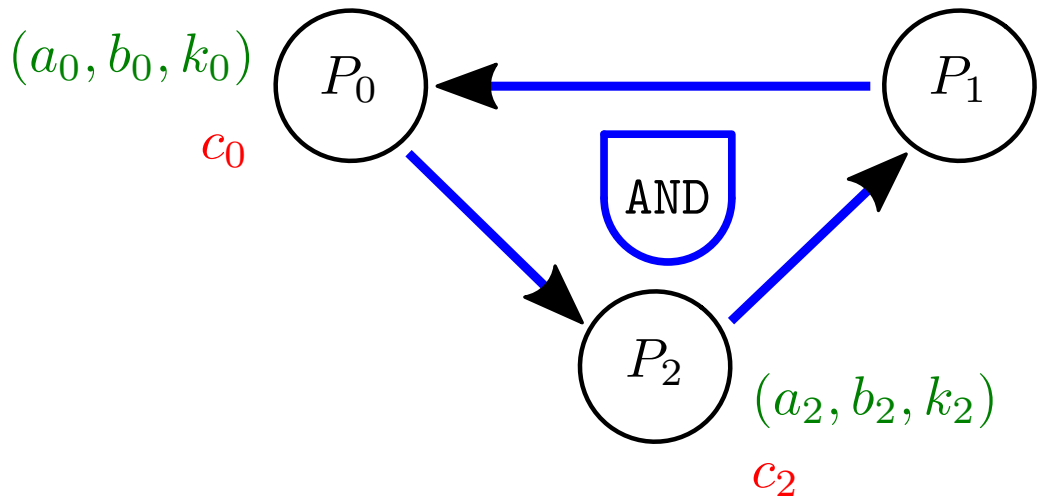
$A = a_0 + a_1 + a_2$
$B = b_0 + b_1 + b_2$
$C = c_0 + c_1 + c_2$

Verifiquesé:

$C = A \texttt{ AND } B$

1. *Shares* de entrada iniciales y bits aleatorios: a_i, b_i, k_i
2. Comunicar: $P_{i-1 \pmod 3}$ recibe (a_i, b_i, k_i)
3. Calcular *shares* de salida: $c_i = a_i b_i + a_i b_{i+1} + a_{i+1} b_i + k_i + k_{i+1}$
4. Comprometerse ■ con todos los valores
5. Revelar vista de dos agentes (e.g., 0 and 2): (a_0, b_0, k_0, c_0) and (a_2, b_2, k_2, c_2)

S3PC en la cabeza



Dados:

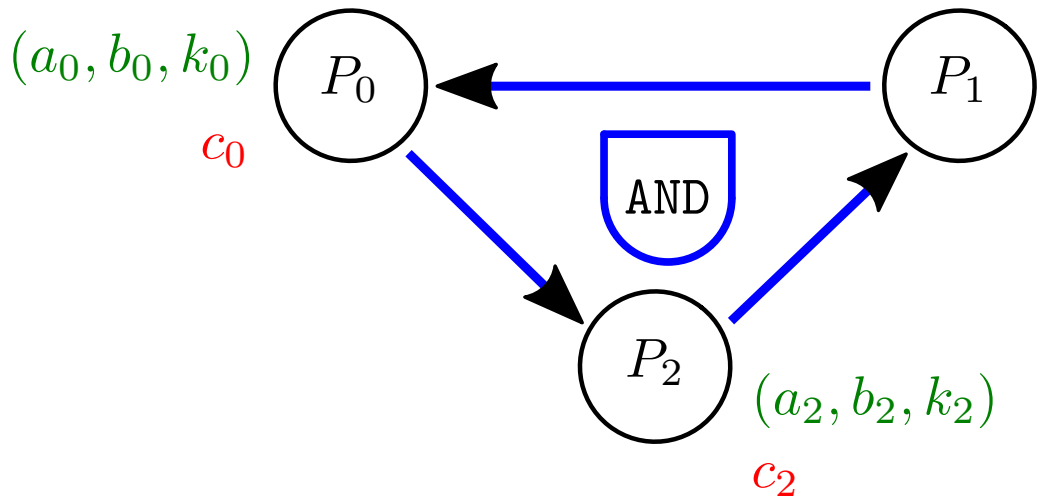
$A = a_0 + a_1 + a_2$
$B = b_0 + b_1 + b_2$
$C = c_0 + c_1 + c_2$

Verifiquesé:

$C = A \texttt{ AND } B$

1. *Shares* de entrada iniciales y bits aleatorios: a_i, b_i, k_i
2. Comunicar: $P_{i-1 \pmod 3}$ recibe (a_i, b_i, k_i)
3. Calcular *shares* de salida: $c_i = a_i b_i + a_i b_{i+1} + a_{i+1} b_i + k_i + k_{i+1}$
4. Comprometerse ■ con todos los valores
5. Revelar vista de dos agentes (e.g., 0 and 2): (a_0, b_0, k_0, c_0) and (a_2, b_2, k_2, c_2)
6. Verificar vista de un agente: $a_2 b_2 + a_2 b_0 + a_0 b_2 + k_2 + k_0 \stackrel{?}{=} c_2$

S3PC en la cabeza



Dados:

$A = a_0 + a_1 + a_2$
$B = b_0 + b_1 + b_2$
$C = c_0 + c_1 + c_2$

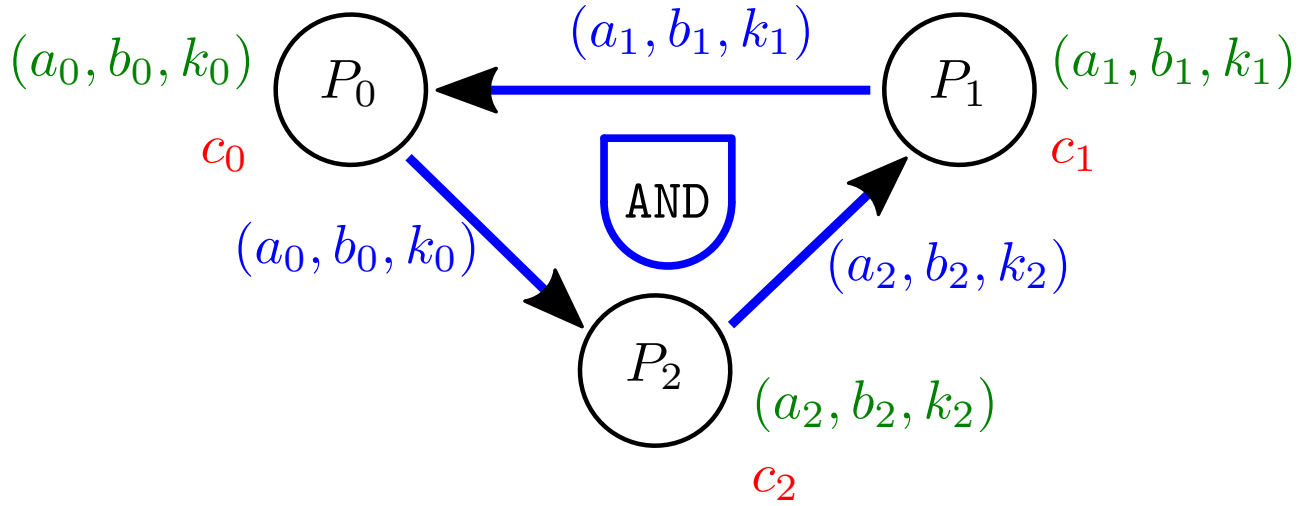
Verifiquesé:

$C = A \texttt{ AND } B$

1. *Shares* de entrada iniciales y bits aleatorios: a_i, b_i, k_i
2. Comunicar: $P_{i-1 \pmod 3}$ recibe (a_i, b_i, k_i)
3. Calcular *shares* de salida: $c_i = a_i b_i + a_i b_{i+1} + a_{i+1} b_i + k_i + k_{i+1}$
4. Comprometerse ■ con todos los valores
5. Revelar vista de dos agentes (e.g., 0 and 2): (a_0, b_0, k_0, c_0) and (a_2, b_2, k_2, c_2)
6. Verificar vista de un agente: $a_2 b_2 + a_2 b_0 + a_0 b_2 + k_2 + k_0 \stackrel{?}{=} c_2$

Nota: esto solo verifica una de las tres *shares* de salida

S3PC en la cabeza



Dados:

$A = a_0 + a_1 + a_2$
$B = b_0 + b_1 + b_2$
$C = c_0 + c_1 + c_2$

Verifiquesé:

$C = A \texttt{ AND } B$

1. *Shares* de entrada iniciales y bits aleatorios: a_i, b_i, k_i
2. Comunicar: $P_{i-1 \pmod 3}$ recibe (a_i, b_i, k_i)
3. Calcular *shares* de salida: $c_i = a_i b_i + a_i b_{i+1} + a_{i+1} b_i + k_i + k_{i+1}$
4. Comprometerse ■ con todos los valores
5. Revelar vista de dos agentes (e.g., 0 and 2): (a_0, b_0, k_0, c_0) and (a_2, b_2, k_2, c_2)
6. Verificar vista de un agente: $a_2 b_2 + a_2 b_0 + a_0 b_2 + k_2 + k_0 \stackrel{?}{=} c_2$

Nota: esto solo verifica una de las tres *shares* de salida

El esquema de la firma Picnic ... en pocas palabras

$$\sigma = \mathsf{NIZKPoK}(x : y = f_u(x); r, c = H(r, pk||m))$$

Leyenda:

- σ: el valor de la firma
- NIZKPoK: prueba no-interactiva de nula divulgación de conocimiento
- $x \equiv sk$: la clave de firma secreta (SignKey)
- $(y = f_u(x), u) \equiv pk$: la clave de verificación pública (VeriKey)
- f: una familia de funciones unidireccionales (Low-MC)
- r: aleatoriedad utilizada en la prueba (Σ)
- c: desafío utilizado en la prueba (Σ)
- m: el mensaje firmado
- H: una función hash

Pauta 4

Observaciones finales

¿Los casos de uso imaginables se ajustan al proceso que se está desarrollando?

- **Queremos:** construir cosas que resulten útiles para lograr funcionalidades múltiples.
- **No queremos:** pasar 10 años en algo y no habilitar algo que ya sabemos que es importante.

Observaciones finales

¿Los casos de uso imaginables se ajustan al proceso que se está desarrollando?

- **Queremos:** construir cosas que resulten útiles para lograr funcionalidades múltiples.
- **No queremos:** pasar 10 años en algo y no habilitar algo que ya sabemos que es importante.

NIST está interesado en el desarrollo criptográfico y la interoperabilidad

- Esto se logra a través de estándares y material de referencia.
- NIST PEC quiere mantenerse al día y apoyar iniciativas externas (como por ejemplo ZKProof).

Observaciones finales

¿Los casos de uso imaginables se ajustan al proceso que se está desarrollando?

- **Queremos:** construir cosas que resulten útiles para lograr funcionalidades múltiples.
- **No queremos:** pasar 10 años en algo y no habilitar algo que ya sabemos que es importante.

NIST está interesado en el desarrollo criptográfico y la interoperabilidad

- Esto se logra a través de estándares y material de referencia.
- NIST PEC quiere mantenerse al día y apoyar iniciativas externas (como por ejemplo ZKProof).

Nos gustaría contar con su colaboración: aplicaciones externas que usan aleatoriedad del Faro y mejoran la privacidad; ...

La prueba del tiempo

¿Cómo serán la privacidad, la seguridad y la criptografía en 70 años?

La prueba del tiempo

¿Cómo serán la privacidad, la seguridad y la criptografía en 70 años?

Photo in 1948 *

Photo in 2018: https://www.nist.gov/sites/default/files/documents/2018/06/15/nist_gaithersburg_master_plan_may_7_2018.pdf

El muro de prueba de piedra NIST (*The NIST Stone Test Wall*): "Construido [en 1948] para estudiar el rendimiento de la piedra sometida a la intemperie. Contiene 2352 muestras individuales de piedra, de las cuales 2032 son piedras nacionales de 47 estados, y 320 son piedras de 16 países extranjeros."

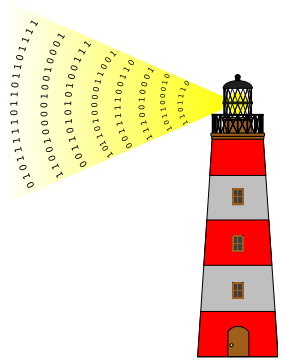
* https://www.nist.gov/el/materials-and-structural-systems-division-73100/nist-stone-wall

- NISTIR 8213: https://doi.org/10.6028/NIST.IR.8213-draft
- Beacon project: https://csrc.nist.gov/projects/interoperable-randomness-beacons

Gracias

- NISTIR 8213: https://doi.org/10.6028/NIST.IR.8213-draft
- Beacon project: https://csrc.nist.gov/projects/interoperable-randomness-beacons

Normas, Investigación y Aplicaciones en Criptografía en NIST



(Algunas notas sobre PEC y PQC)

luis.brandao@nist.gov; rene.peralta@nist.gov

Presentación en

7 de Octubre, 2019 @ Santiago, Chile

Lista de transparencias