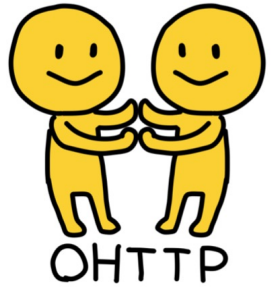
Internet Week
2023

PKI(application)の
モデル変遷

2023/11/21
NTT社会情報研究所 奥田 哲矢

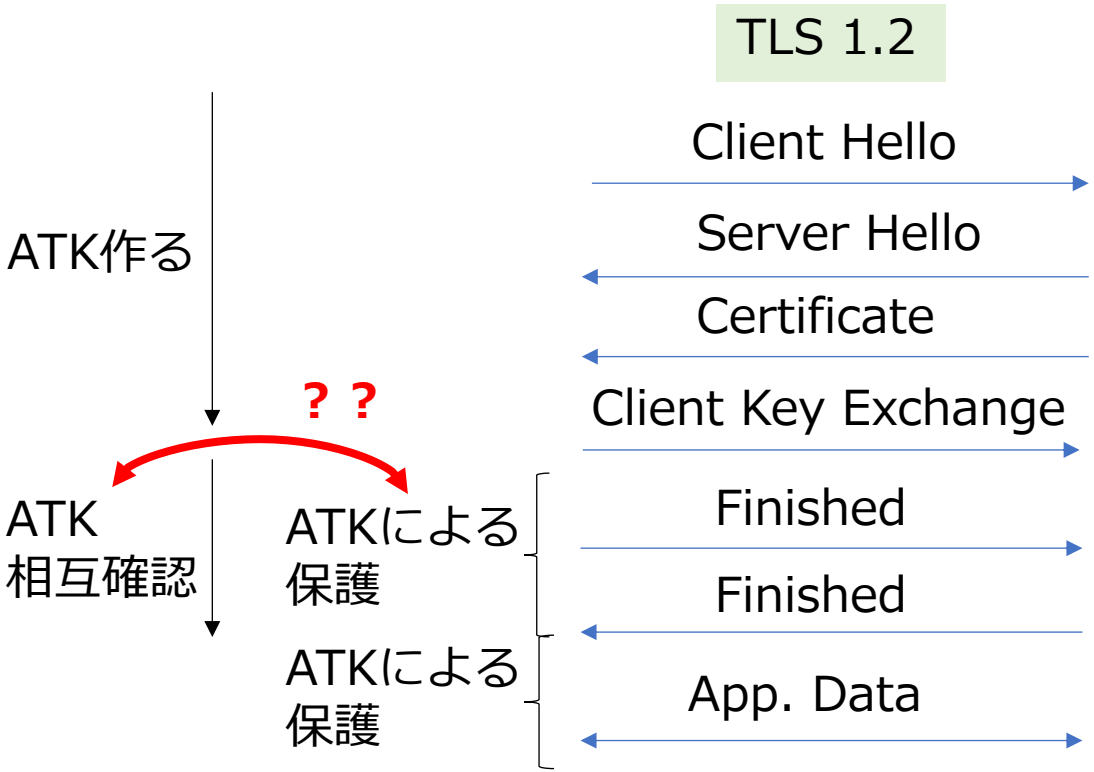


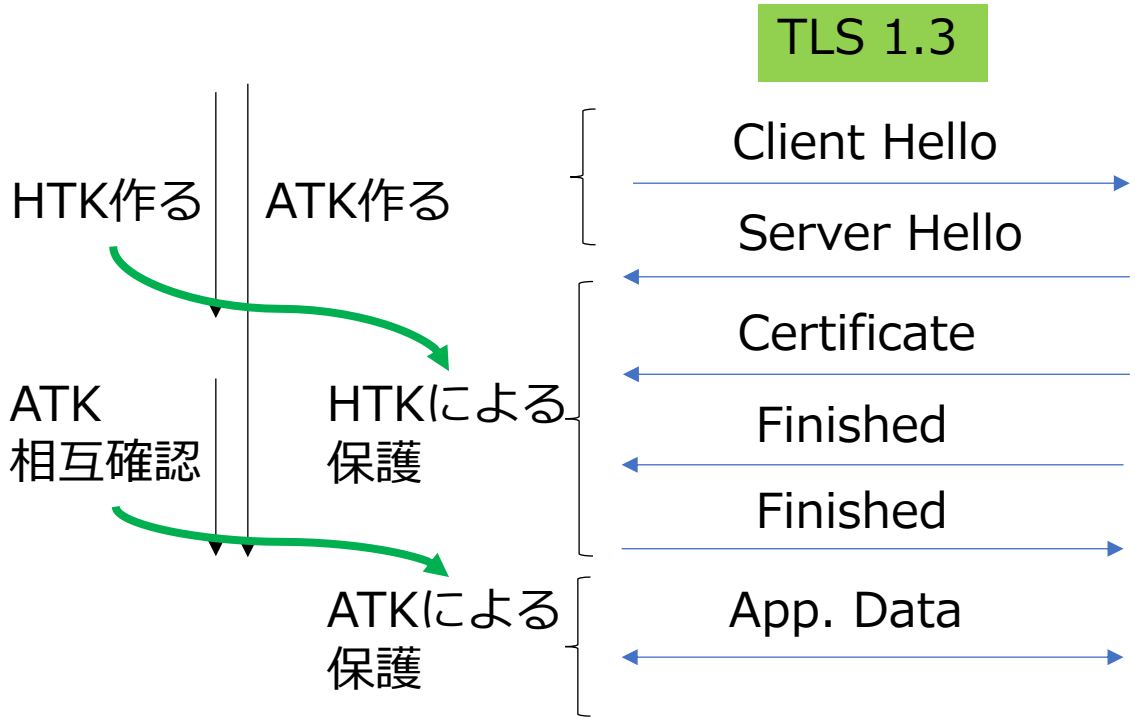
TLS 1.3 のHandShake暗号化

・TLS1.2はATKを作る時点と使う時点に不整合があった。

→TLS1.3はHTK, ATKに鍵の目的を区別して整理した。
HandShakeをなるべく暗号化した（プライバシ保護）

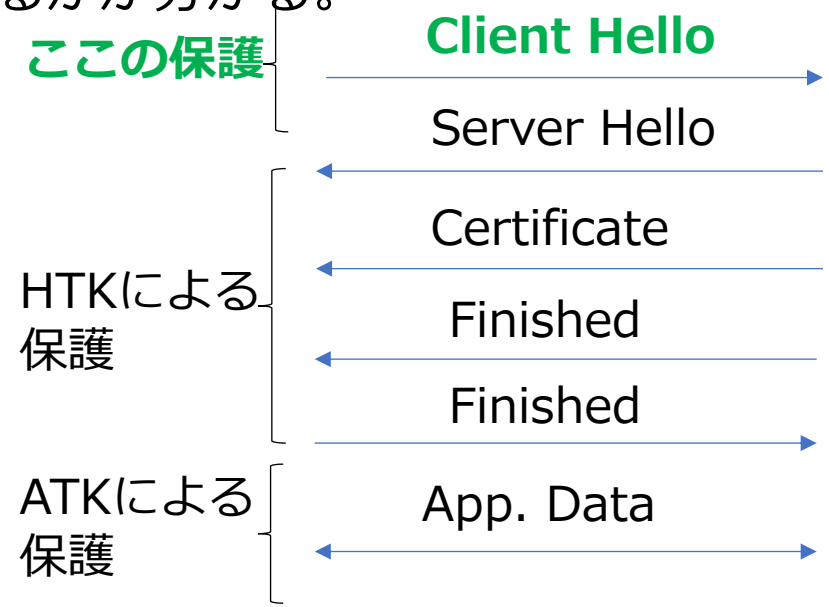
ATK:Application Traffic Key(Secret)
HTK:Application Traffic Key(Secret)





TLS / ECH (Encrypted Client Hello)

・TLSはプライバシー強化を目指して進化中
(cf. ECHのDraftに将来を見て”TLS1.3 or higher”の記載あり)
・最初のClient Hello(CH)は暗号化されない。CHはTLSで保護すべき対象データではないが、
ここに含まれる属性情報からユーザが特定される可能性を懸念。
・元々、ESNI(Encrypted Server Name Indicator)が提案されてきた。
SNIは、CDN等の中継サーバに接続対象サーバを指定する仕組み。
CHに含まれるSNIを見れば、ユーザがどのサービスを利用しているかが分かる。
→ECHは、保護対象をClient Hello全体に拡大する提案。



TLS / ECH 仕様
[draft-ietf-tls-esni-17] TLS Encrypted Client Hello

TLS / ECH (Encrypted Client Hello) & HPKE(Hybrid Public Key Encryption)

・最初のClient Helloを暗号化する。

→そもそもTLS Handshakeが終わってないのにどうやって？

→HPKE(Hybrid Public Key Encryption)：

KEM/DEM構成のハイブリッド暗号化(※耐量子のそれではない)

KEM/KDF/AEADを部品として取替可能なフレームワーク

CFRG(IETFの暗号部門)で議論、INRIAの研究者の方々の貢献もあり、

KEM/DEM構成のハイブリッド暗号化の次の標準となる見込みが高い。

（OpenSSLでも次のver.3.2で実装される様子）

(cf. DHIES,ECIES)

[RFC 9180] Hybrid Public Key Encryption
R. Barnes Cisco, K. Bhargavan B. Lipp Inria, C. Wood Cloudflare, February 2022.

安全性評価の論文も幾つか公開されている

Alwen, J., **Blanchet, B.**, Hauck, E., Kiltz, E., Lipp, B., and D. Riepel, "Analysing the HPKE Standard", EUROCPYRT 2021.

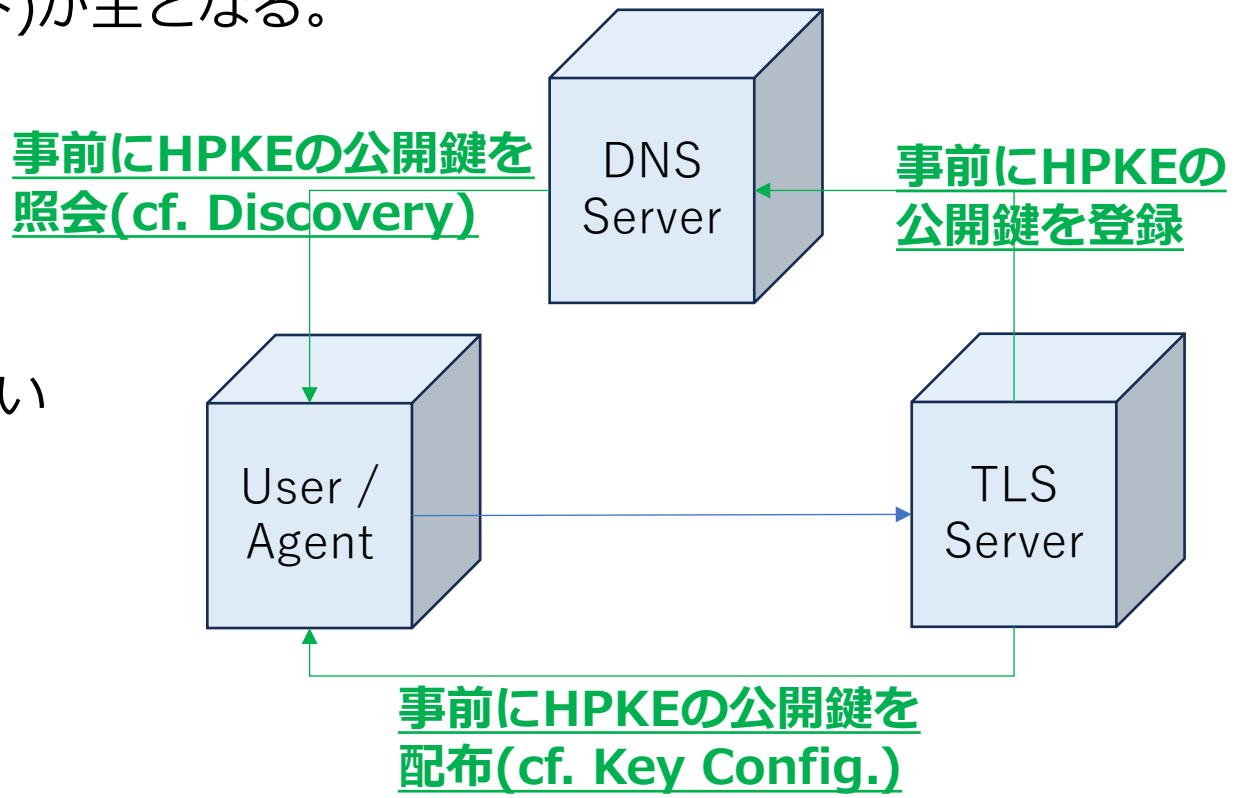
B. Lipp, "An Analysis of Hybrid Public Key Encryption", IACR ePrint.

TLS / ECH & HPKE (※PKI的には)

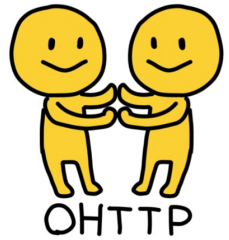
・KEM/DEM構成のハイブリッド暗号化
→そもそも公開鍵をどうやって入手するのか？
→おそらくDNS(HTTPS(SVCB)リソースレコード)が主となる。
DNSに公開鍵自体を登録するか、
TLS ServerからKey Config.を配布する。

・TLS 1.3 0-RTT / Early Data と同様、
基本モードでは幾つかの安全性要件を満たさない
(cf. PFS, KCI, 冪等) とは言え、
応用先の限定や鍵の運用方針を上手く決めて、
TLS / ECH や OHTTP(※後述) で活用が進む。

※DNSのプロの皆さんが多い場なので、
ぜひ認識違いなどあれば後学のためご教示ください。



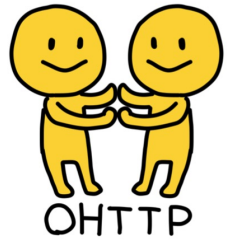
OHTTP & ODoH



・OHTTP(Oblivious HTTP)
・ODoH(Oblivious DNS over HTTPS)
最近のIETFプロトコルでは、
プライバシーへの影響がよく議論されている。
→その先駆けはDoH(DNS over HTTPS)（※筆者私見）
Always EncryptionからのDNSクエリの通信路の秘匿。
これで、通信路上の誰かに対しては、
　ユーザが利用するサービスを秘匿出来たが、
例えばDNSサーバはユーザも／利用するサービスも／知っているか。

※DNSのプロの皆さんが多い場なので、
ぜひ認識違いなどあれば後学のためご教示ください。

OHTTP & ODoH Req.



- OHTTP(Oblivious HTTP)
- ODoH(Oblivious DNS over HTTPS)

要求元と要求／応答内容を同時に知る
エンティティが居ないように設計している

事前にHPKEの公開鍵を
開示(cf. Discovery)

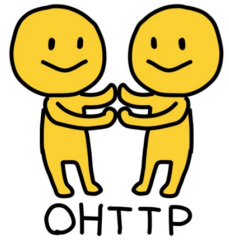
OHTTP req.
- 要求元/metadata
- 要求先/接続先
- 要求内容をHPKE (KEM/KDF/AEAD)で **暗号化**

OHTTP req.
- 要求元/metadata
- 要求先/接続先
- 要求内容はHPKE (KEM/KDF/AEAD)で暗号化されている 秘匿

OHTTP req.
- 要求元/metadata 秘匿
- 要求先/接続先
- 要求内容をHPKE (KEM/KDF/AEAD)で **復号**

※要求元の絞り込みに使われ得る
メタデータを合わせて秘匿する。

OHTTP & ODoH Res.



- OHTTP(Oblivious HTTP)
- ODoH(Oblivious DNS over HTTPS)

要求元と要求／応答内容を同時に知る
エンティティが居ないように設計している

OHTTP res.
- 要求元/metadata
- 要求先/接続先
- 応答内容をHPKE (KDF/AEAD)で **復号**

OHTTP res.
- 要求元/metadata
- 要求先/接続先
- 応答内容はHPKE (KDF/AEAD)で

暗号化されている 秘匿

OHTTP res.
- 要求元/metadata 秘匿
- 要求先/接続先
- 応答内容をHPKE (KDF/AEAD)で **暗号化**

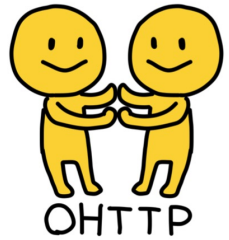
OHTTP res.
- 要求元/metadata 秘匿
- 要求先/接続先
- 応答内容は平文

※要求元の絞り込みに使われ得る
メタデータを合わせて秘匿する。

※DNSのプロの皆さんが多い場なので、
ぜひ認識違いなどあれば後学のためご教示ください。

OHTTP & ODoH

要求元と要求／応答内容を同時に知る
エンティティが居ないように設計している

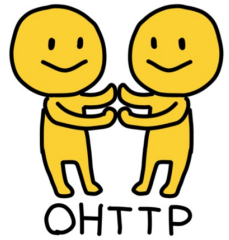


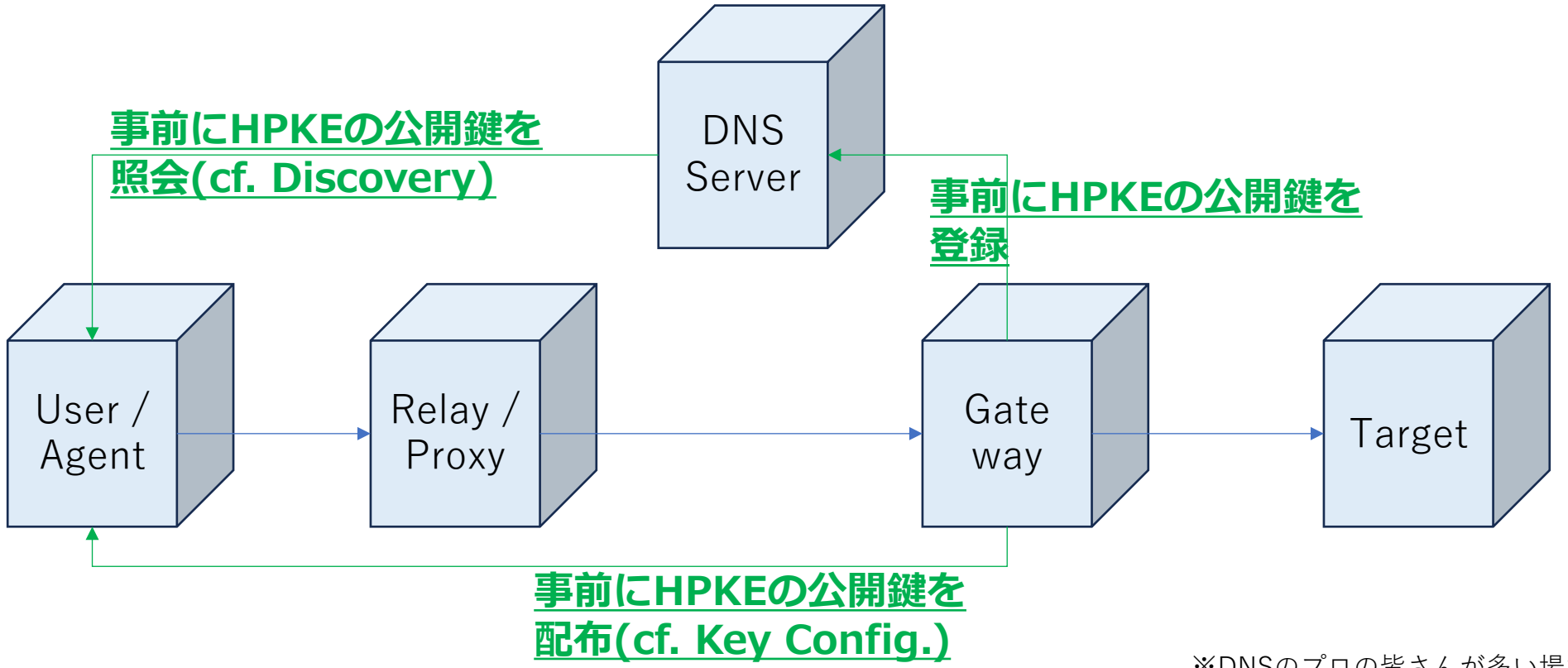
ユーザは知ってるが
要求内容は知らない

ユーザは知らないが（※仮名）
要求内容は知ってる

・ユーザと利用サービス／要求内容の関連付けを区切る最初の目的は達成。
　でも、Gateway＆Targetは仮名(connection/key id)は知っている。
　一回のHPKEの後、それらのid(state)で個人が絞り込みされないだろうか。
→HPKEの実行頻度については、ストリーミング利用を中心に議論継続中。
→基本的には、DNSクエリやテレメトリデータ送信など、
　”一回限り”(one-time/one-round)の利用に相性が良い仕組み。

OHTTP & ODoH (※PKI的には)

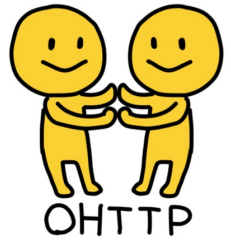




※DNSのプロの皆さんが多い場なので、
ぜひ認識違いなどあれば後学のためご教示ください。

・HPKEに必要な公開鍵配布の仕組みは、
おそらくDNS(HTTPS(SVCB)リソースレコード)が主となる。
DNSに公開鍵自体を登録するか、GatewayからKey Config.を配布する。

OHTTP & ODoH 仕様議論



◇OHTTP仕様策定に関する状況

・議論：stateとunlinkabilityに関する議論がある（という認識）

state(鍵やコネクションのIDなど)を維持すると、

一般にunlinkabilityを充足しなくなる。

(HPKEによる)更新頻度をどう設定するか。

・その他の疑問：Relay/Proxyにおける対応テーブル(実名⇄仮名)の作り方の任意性

Relay/Proxyは実名ユーザの利用サービスを知り得るか？

（TLS ECHと組み合わせる等の対応が必要となる想定か）

Gatewayは仮名ユーザを利用サービスで絞り込み得るか？

（GatewayがTarget/利用サービスと同じ管理元であれば問題ないか）

※DNSのプロの皆さんが多い場なので、
ぜひ認識違いなどあれば後学のためご教示ください。

[draft-ietf-ohai-ohttp-10] Oblivious HTTP
※OHTTPのWG、2021/11のIETF112~117を聴講して
下記のスライドの構成図が最も分かり易い印象
Oblivious HTTP (slides-114-ohai-oblivious-http)
[RFC 9230]Oblivious DNS over HTTPS (未読)

免責事項

・まだまだ進行中＆勉強中のプロジェクトのため、
　技術的な内容や記載に誤りがあればご指摘頂けますと幸いです。