NEXT GENERATION INTERNET

Anonymity

Christian Grothoff

23.05.2025

Learning Objectives

What is Anonymity?

How can we achive anonymity on the Internet?

How does onion routing work?

Advanced Cryptographic Primitives

Secure Multiparty Computation

Part I: What is Anonymity?

Motivation

Suppose Alice and Bob communicate using encryption.

What can Eve still learn here?

How much does TLS leak?

“We present a traffic analysis attack against over 6000 webpages spanning the HTTPS deployments of 10 widely used, industry-leading websites in areas such as healthcare, finance, legal services and streaming video. Our attack **identifies individual pages** in the same website with 89% accuracy, exposing personal details including **medical conditions**, financial and **legal affairs** and **sexual orientation**. We examine evaluation methodology and reveal accuracy variations as large as 18% caused by assumptions affecting caching and cookies.” [15]

Anonymity definitions

Merriam-Webster:

1. not named or identified: “an anonymous author”, “they wish to remain anonymous”
2. of unknown authorship or origin: “an anonymous tip”
3. lacking individuality, distinction, or recognizability: “the anonymous faces in the crowd”, “the gray anonymous streets” – William Styron

Anonymity definitions

Andreas Pfitzmann et. al.:

> “Anonymity is the state of being not identifiable within a set of subjects, the anonymity set.”

Anonymity definitions

Andreas Pfitzmann et. al.:

"Anonymity is the state of being not identifiable within a set of subjects, the anonymity set."

EFF:

"Instead of using their true names to communicate, (...) people choose to speak using pseudonyms (assumed names) or anonymously (no name at all)."

Anonymity definitions

Andreas Pfitzmann et. al.:

"Anonymity is the state of being not identifiable within a set of subjects, the anonymity set."

EFF:

"Instead of using their true names to communicate, (...) people choose to speak using pseudonyms (assumed names) or anonymously (no name at all)."

Our definition:

A user's action is anonymous if the adversary cannot link the action to the user's identity

The user's identity

includes personally identifiable information, such as:

- real name
- fingerprint
- passport number
- IP address
- MAC address
- login name
- ...

Actions

include:

- Internet access
- speach
- participation in demonstration
- purchase in a store
- walking across the street
- ...

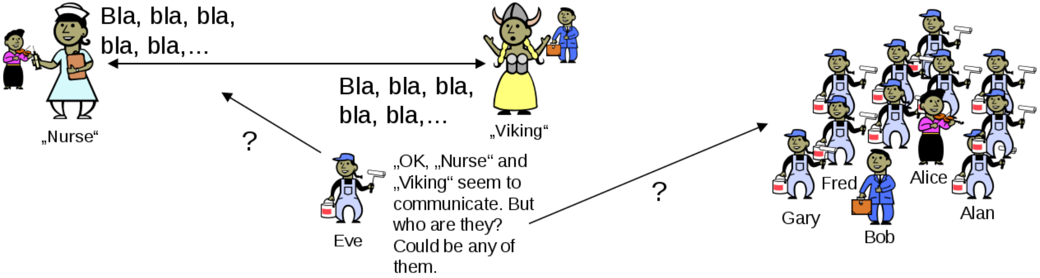
Anonymity: Terminology

- Sender Anonymity: The initiator of a message is anonymous. However, there may be a path back to the initiator.

- Receiver Anonymity: The receiver of a message is anonymous.

Pseudonymity

A pseudonym is an alternative name for an entity in the system.



A pseudonym can be tracked. We can observe its behaviour, but we should not learn the identity of who is behind it.

Evaluating anonymity

How much anonymity does a given system provide?

- Number of known attacks?
- Lowest complexity of successful attacks?
- Number of users?
- Information leaked through messages and maintenance procedures?

Anonymity: Basics

- **Anonymity Set** is the set of suspects
- Attacker computes a **probability distribution** describing the likelyhood of each participant to be the responsible party.
- Anonymity is the stronger, the larger the anonymity set and the more evenly distributed the subjects within that set are.

Anonymity metric: Anonymity Set Size

Let $\mathcal{U}$ be the attacker's probability distribution and $p_u = \mathcal{U}(u)$ describing the probability that user $u \in \Psi$ is responsible.

$$ASS := \sum_{\substack{u \in \Psi \\ p_u > 0}} 1 \qquad (1)$$

Large anonymity sets

Examples of large anonymity sets:

- Any human

Large anonymity sets

Examples of large anonymity sets:

- Any human
- Any human with Internet access

Large anonymity sets

Examples of large anonymity sets:

- Any human
- Any human with Internet access
- Any human speaking German

Large anonymity sets

Examples of large anonymity sets:

- Any human
- Any human with Internet access
- Any human speaking German
- Any human speaking German with Internet access awake at 3am CEST

Anonymity metric: Maximum Likelihood

Let $\mathcal{U}$ be the attacker's probability distribution describing the probability that user $u \in \Psi$ is responsible.

$$ML := \max_{u \in \Psi} p_u \qquad (2)$$

Anonymity metric: Maximum Likelihood

- For successful criminal prosecution in the US, the law requires *ML close* to 1 ("beyond reasonable doubt")
- For successful civil prosecution in the US, the law requires $ML > \frac{1}{2}$ ("more likely than not")
- For a given anonymity set, the best anonymity is achieved if

$$ML = \frac{1}{ASS} \tag{3}$$

Anonymity metric: Entropy

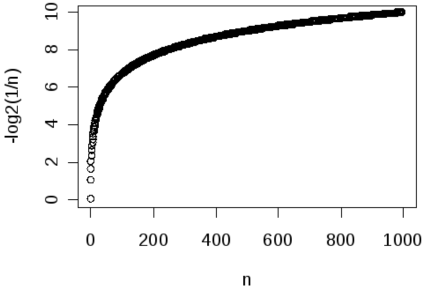
Let $\mathcal{U}$ be the attacker's probability distribution describing the probability that user $u \in \Psi$ is responsible. Define the effective size S of the anonymity distribution $\mathcal{U}$ to be:

$$S := -\sum_{u \in \Psi} p_u \log_2 p_u \tag{4}$$

where $p_u = \mathcal{U}(u)$.

Interpretation of entropy

$$S = -\sum_{u \in \Psi} p_u \log_2 p_u \tag{5}$$



Entropy calculation example

Suppose we have 101 suspects including Bob. Furthermore, suppose for Bob the attacker has a probability of 0.9 and for all the 100 other suspects the probability is 0.001.

What is S?

Entropy calculation example

Suppose we have 101 suspects including Bob. Furthermore, suppose for Bob the attacker has a probability of 0.9 and for all the 100 other suspects the probability is 0.001.

What is S?

- For 101 nodes $H_{max} = 6.7$
- $$\begin{aligned} S &= -\frac{100 \cdot \log_2 0.001}{1000} - \frac{9 \cdot \log_2 0.9}{10} && (6)\\ &\approx 0.9965 + 0.1368 && (7)\\ &= 1.133... && (8) \end{aligned}$$

Attacks to avoid

Hopeless situations include:

- All nodes collaborate against the user
- All directly adjacent nodes collaborate
- All non-collaborating adjacent nodes are made unreachable from the user
- The user is required to prove her innocence

Economics & Anonymity

There are hard issues in *the Economics of Anonymity* [1]:

- Providing anonymity services has economic disincentives (DoS, legal liability)
- Anonymity requires introducing inefficiencies!

$\Rightarrow$ Who pays for that?

Economics & Anonymity

There are hard issues in *the Economics of Anonymity* [1]:

- Providing anonymity services has economic disincentives (DoS, legal liability)
- Anonymity requires introducing inefficiencies!

⇒ Who pays for that?

Anonymity Trilemma

The Anonymity Trilemma [7] states that given the objectives of:

- Strong anonymity
- Low bandwidth overhead
- Low latency

... one can only have two of the three.

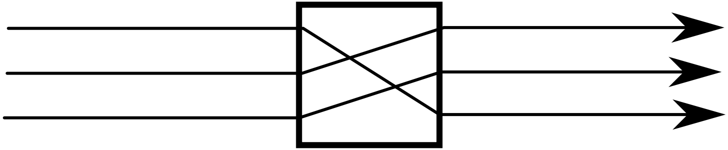
Part II: How to achieve anonymity?

Anonymity: Dining Cryptographers

"Three cryptographers are sitting down to dinner. The waiter informs them that the bill will be paid anonymously. One of the cryptographers maybe paying for dinner, or it might be the NSA. The three cryptographers respect each other's right to make an anonymous payment, but they wonder if the NSA is paying." – David Chaum

Mixing

David Chaum's mix (1981) and cascades of mixes are the traditional basis for destroying linkability:

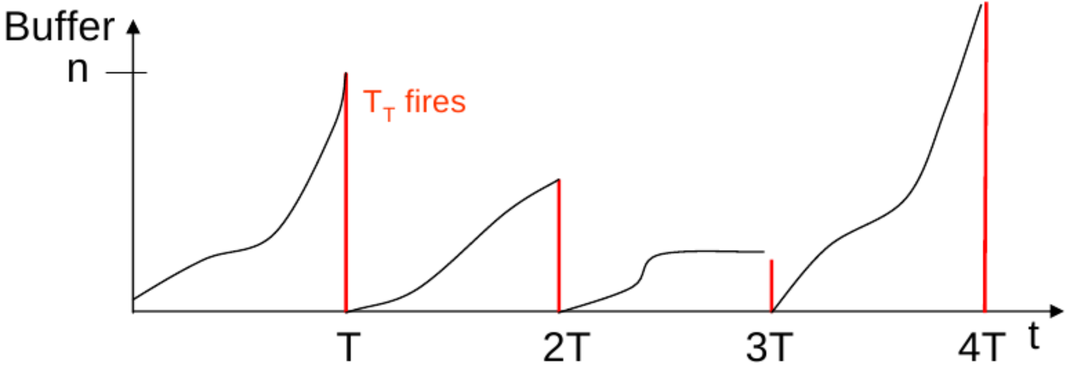


Mixing

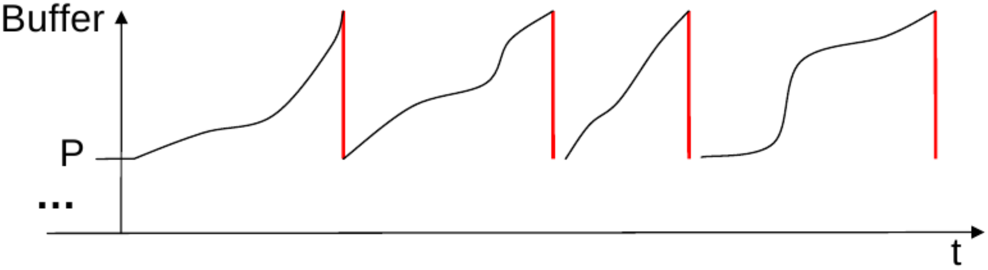
David Chaum's mix (1981) and cascades of mixes are the traditional basis for destroying linkability:

Threshold Mix

Timed Mix



Pool mix



Mixminion

G. Danezis, R. Dingledine, D. Hopwood and N. Mathewson describe Mixminion [5]:

- builds on the idea of remailers: Mixes for E-mail
- possibility to reply
- directory servers to evaluate participating remailers (reputation system)
- exit policies
- dummy traffic

Mixminion: key ideas

When a message traverses mixminion, each node must decrypt the message using its (ephemeral) private key.

The key idea behind **replies** is splitting the path into two legs:

- the first half is chosen by the responder to hide the responder identity
- the second half was communicated by the receiver to hide the receiver identity
- a crossover-node in the middle is used to switch the headers specifying the path

Mixminion: replay?

Replay attacks were an issue in previous mixnet implementations.

- Mixes are vulnerable to replay attacks
- Mixminion: servers keep hash of previously processed messages until the server key is rotated

⇒ Bounded amount of state in the server, no possibility for replay attack due to key rotation

Mixminion: Directory Servers

- Inform users about servers
- Probe servers for reliability
- Allow a partitioning attack unless the user always queries all directory servers for everything

Mixminion: Nymservers

- Nymservers keep list of use-once reply blocks for a user
- Vulnerable to DoS attacks (deplete reply blocks)
- Nymservers could also store mail (use one reply block for many messages).

Mixminion: obvious problems

- no benefits for running a mixmailer for the operator
- quite a bit of public key cryptography
- trustworthiness of directory servers questionable
- servers must keep significant (but bounded) amount of state
- limited to E-mail (high latency)

Mixminion: open problems

- exit nodes are fair game for legal actions
- no accounting to defend against abuse / DoS attacks
- statistical correlation of entities communicating over time possible (observe participation)

⇒ bridging between an anonymous network and a traditional protocol is difficult

Subsequent remailer research has focused on improving the cryptography [6, 16] and integrating economic incentives for operators [8].

`https://nymtech.com/` and `https://github.com/katzenpost/katzenpost` are modern examples.

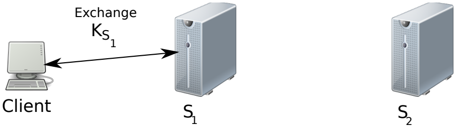
Part III: Onion Routing

Onion Routing

- Multiple mix servers
- Path of mix servers chosen by initiator
- Chosen mix servers create “circuit”
  - Initiator contacts first server S_1, sets up symmetric key K_{S_1}
  - Then asks first server to connect to second server S_2; through this connection sets up symmetric key with second server K_{S_2}
  - ...
  - Repeat with server S_i until circuit of desired length n constructed

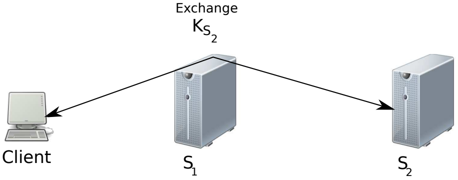
Onion Routing Example

- Client sets up symmetric key K_{S_1} with server S_1



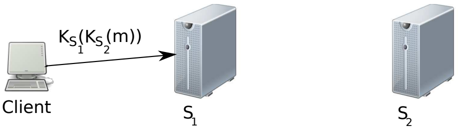
Onion Routing Example

- Via S_1, the client sets up symmetric key K_{S_2} with server S_2



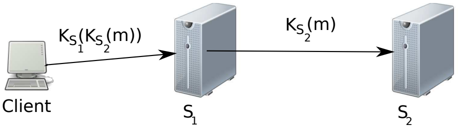
Onion Routing Example

- Client encrypts m as $E(K_{S_1}, E(K_{S_2}, (m)))$ and sends to S_1



Onion Routing Example

- Server S_1 decrypts and forwards $E(K_{S_2},(m))$ to S_2.



- S_2 decrypts, revealing m.

Tor [9]

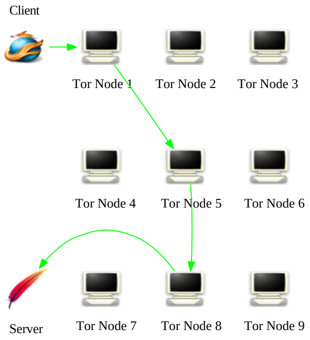
- Tor is a P2P network of **low-latency** mixes which use onion routing to provide anonymous communication between parties on the Internet.
- Tor works for any TCP-based protocol and is designed for interactive traffic (https, ssh, etc.)
- TCP traffic enters the Tor network via a SOCKS proxy
- **Common usage:** client anonymity for Web browsing

Tor - How it Works

- ”Directory Servers“ store list of participating servers
  - Contact information, public keys, statistics
  - Directory servers are replicated for security
- Clients choose servers randomly with bias towards high BW/uptime
- Clients build long lived Onion routes ”circuits“ using these servers
- Circuits are bi-directional
- Circuits are of length three

Tor - How it Works - Example

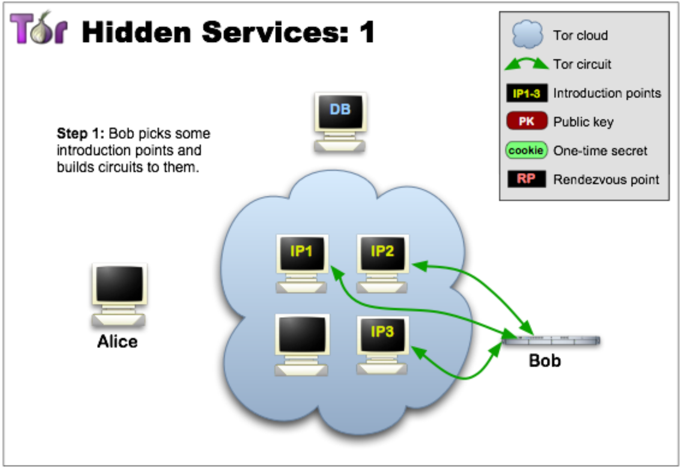
- Example of Tor client circuit



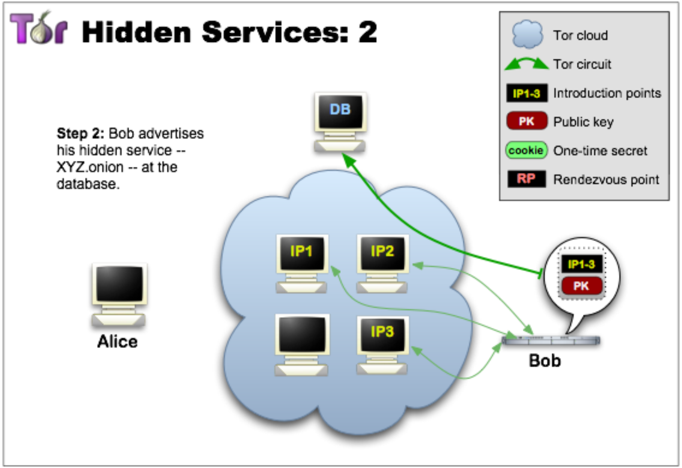
Hidden Services in Tor

- Hidden services allow Tor servers to receive incoming connections anonymously
- Can provide access to services available *only* via Tor
  - Web, IRC, etc.
  - For example, host a website without your ISP knowing

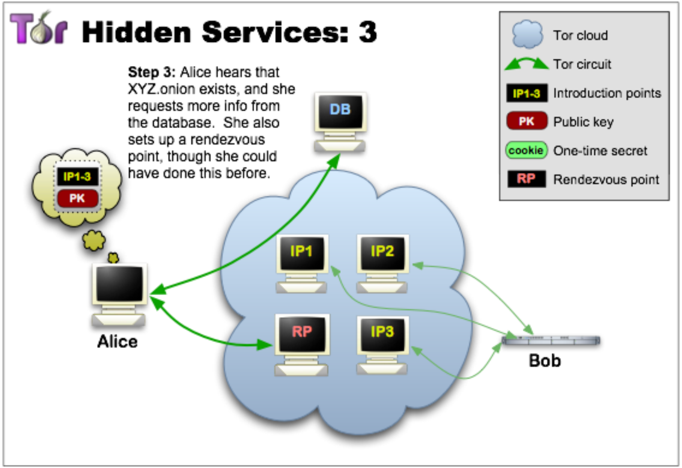
Hidden Services Example 1



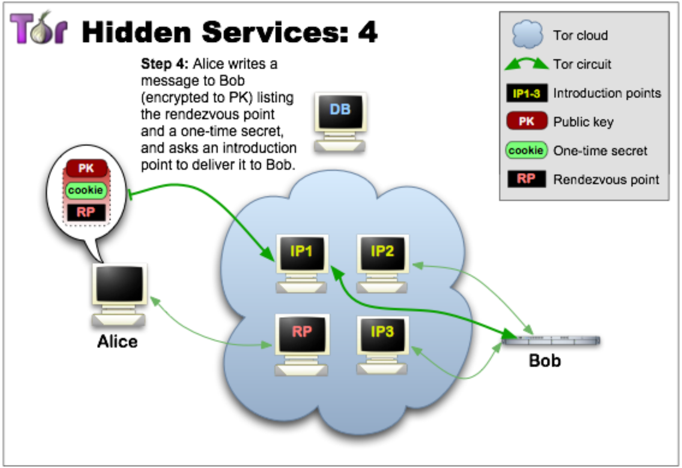
Hidden Services Example 2



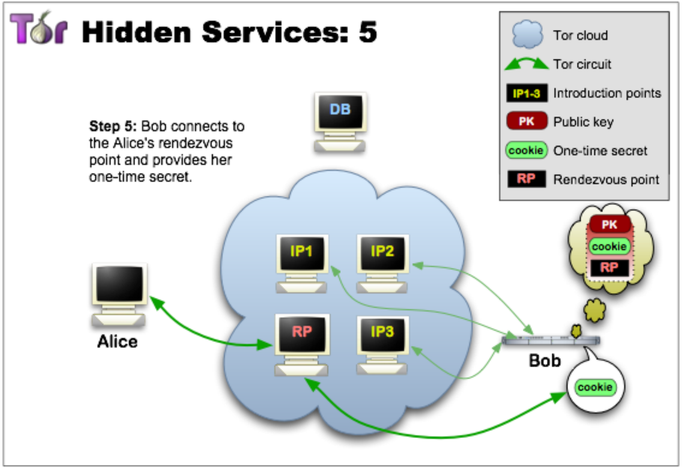
Hidden Services Example 3



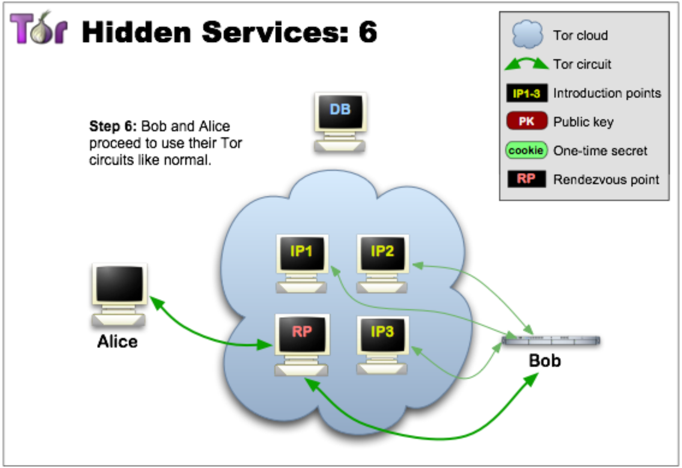
Hidden Services Example 4



Hidden Services Example 5



Hidden Services Example 6



Types of Attacks on Tor

- Exit relay snooping
- Website fingerprinting
- Traffic analysis
- Intersection attacks
- DoS [10]

An avoidiable (but historically common) issue are badly configured hidden services that directly expose critical information about the operator by accident over the application protocol.

Part IV: Advanced Cryptographic Primitives

Homomorphic Encryption

$$E(x_1 \oplus x_2) = E(x_1) \otimes E(x_2) \qquad (9)$$

Multiplicative Homomorphism: RSA & ElGamal

- Unpadded RSA (multiplicative):

$$E(x_1) \cdot E(x_2) = x_1^e x_2^e = E(x_1 \cdot x_2) \quad (10)$$

- ElGamal:

$$\begin{aligned} E(x_1) \cdot E(x_2) &= (g^{r_1}, x_1 \cdot h^{r_1})(g^{r_2}, x_2 \cdot h^{r_2}) && (11)\\ &= (g^{r_1+r_2}), (x_1 \cdot x_2)h^{r_1+r_2}) && (12)\\ &= E(x_1 \cdot x_2) && (13) \end{aligned}$$

Additive Homomorphism: Paillier

$$E_K(m) := g^m \cdot r^n \mod n^2, \tag{14}$$

$$D_K(c) := \frac{(c^\lambda \mod n^2) - 1}{n} \cdot \mu \mod n \tag{15}$$

where the public key $K = (n, g)$, m is the plaintext, c the ciphertext, n the product of $p, q \in \mathbb{P}$ of equal length, and $g \in \mathbb{Z}^*_{n^2}$. In Paillier, the private key is (λ, μ), which is computed from p and q as follows:

$$\lambda := \text{lcm}(p - 1, q - 1), \tag{16}$$

$$\mu := \left(\frac{(g^\lambda \mod n^2) - 1}{n} \right)^{-1} \mod n. \tag{17}$$

Paillier offers additive homomorphic public-key encryption, that is:

$$E_K(a) \oplus E_K(b) \equiv E_K(a + b) \tag{18}$$

Fully homomorphic encryption

Additive:

$$E(A) \oplus E(B) = E(A + B) \tag{19}$$

and multiplicative:

$$E(A) \otimes E(B) = E(A \cdot B) \tag{20}$$

Known cryptosystems: Brakerski-Gentry-Vaikuntanathan (BGV), NTRU, Gentry-Sahai-Waters (GSW).

Pairing-based cryptography

Let G_1, G_2 be two additive cyclic groups of prime order q, and G_T another cyclic group of order q (written multiplicatively). A pairing is an efficiently computable map e:

$$e : G_1 \times G_2 \rightarrow G_T \quad (21)$$

which satisfies $e \neq 1$ and bilinearity:

$$\forall_{a,b \in F_q^*}, \ \forall_{P \in G_1, Q \in G_2} : \ e(aP, bQ) = e(P, Q)^{ab} \quad (22)$$

Examples: Weil pairing, Tate pairing.

Hardness assumption

Computational Diffie Hellman:

$$g, g^x, g^y \Rightarrow g^{xy} \tag{23}$$

remains hard on G even given e.

Boneh-Lynn-Sacham (BLS) signatures [4]

Key generation:

Pick random $x \in \mathbb{Z}_q$

Signing:

$\sigma := h^x$ where $h := H(m)$

Verification:

Given public key g^x:

$$e(\sigma, g) = e(h, g^x) \tag{24}$$

Boneh-Lynn-Sacham (BLS) signatures [4]

Key generation:

Pick random $x \in \mathbb{Z}_q$

Signing:

$\sigma := h^x$ where $h := H(m)$

Verification:

Given public key g^x:

$$e(\sigma, g) = e(h, g^x) \tag{24}$$

Why:

$$e(\sigma, g) = e(h, g)^x = e(h, g^x) \tag{25}$$

due to bilinearity.

Fun with BLS

Given signature $\langle \sigma, g^x \rangle$ on message h, we can *blind* the signature and public key g^x:

$$e(\sigma^b, g) = e(h, g)^{xb} = e(h, g^{xb}) \tag{26}$$

Thus σ^b is a valid signature for the *derived* public key $(g^x)^b$ with blinding value $b \in \mathbb{Z}_q$.

Part V: Secure Multiparty Computation

Secure Multiparty Computation (SMC)

- Alice und Bob haben private Daten a_i and b_i.
- Alice und Bob führen ein Protokoll aus und berechnen gemeinsam $f(a_i, b_i)$.
- Nur einer von beiden lernt das Ergebnis (i.d.R.)

Adversary models

Honest but curious

Dishonest and curious

Secure Multiparty Computation: Scalar Product

We want to calculate

$$\sum_i a_i b_i \tag{27}$$

- Original idea by Ioannids et al. in 2002 [12] (use: $(a-b)^2 = a^2 - 2ab + b^2$)
- Refined by Amirbekyan et al. in 2007 (corrected math) [2]

SMC (ECC Version)[1]

Let Alice's secret value be $a \in \mathbb{Z}$. Alice sends to Bob $(g_i, h_i) = (g^{r_i}, g^{r_i a + a_i})$ with random values r_i for $i \in M$.
Bob answers with:

$$\left(\prod_{i \in M} g_i^{b_i}, \prod_{i \in M} h_i^{b_i}\right) = \left(\prod_{i \in M} g_i^{b_i}, \left(\prod_{i \in M} g_i^{b_i}\right)^a g^{\sum_{i \in M} a_i b_i}\right)$$

Alice can then calculate:

$$\left(\prod_{i \in M} g_i^{b_i}\right)^{-a} \cdot \left(\prod_{i \in M} g_i^{b_i}\right)^a \cdot g^{\sum_{i \in M} a_i b_i} = g^{\sum_{i \in M} a_i b_i}.$$

Assuming $\sum_{i \in M} a_i b_i$ is sufficiently small, then Alice can compute the scalaproduct by solving the DLP.

[1] Joint work with Tanja Lange

References I

- Alessandro Acquisti, Roger Dingledine, and Paul Syverson.
  On the economics of anonymity.
  In Rebecca N. Wright, editor, *Financial Cryptography*, pages 84–102, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.
- Artak Amirbekyan and Vladimir Estivill-castro.
  A new efficient privacypreserving scalar product protocol.
  In *in Proc. of AusDM '07*, pages 209–214.

References II

- Daniel Arp, Fabian Yamaguchi, and Konrad Rieck.
  Torben: A practical side-channel attack for deanonymizing tor communication.
  In *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, pages 597–602, 2015.
- Dan Boneh, Ben Lynn, and Hovav Shacham.
  Short signatures from the weil pairing.
  In *Advances in Cryptology – ASIACRYPT '01, LNCS*, pages 514–532. Springer, 2001.

References III

- George Danezis, Roger Dingledine, and Nick Mathewson.
  Mixminion: Design of a type iii anonymous remailer protocol.
  In *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, SP '03, 2003.

- George Danezis and Ian Goldberg.
  Sphinx: A compact and provably secure mix format.
  In *2009 30th IEEE Symposium on Security and Privacy*, pages 269–282. IEEE, 2009.

References IV

- Debajyoti Das, Sebastian Meiser, Esfandiar Mohammadi, and Aniket Kate.
  Anonymity trilemma: Strong anonymity, low bandwidth overhead, low latency - choose two.
  In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 108–126, 2018.
- Claudia Diaz, Harry Halpin, and Aggelos Kiayias.
  The nym network.
  2021.

References V

- Roger Dingledine, Nick Mathewson, and Paul Syverson.
  Tor: the second-generation onion router.
  In *Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13*, SSYM'04, page 21, USA, 2004. USENIX Association.

- Nathan S Evans, Roger Dingledine, and Christian Grothoff.
  A practical congestion attack on tor using long paths.
  In *USENIX Security Symposium*, pages 33–50, 2009.

References VI

Alfonso Iacovazzi, Daniel Frassinelli, and Yuval Elovici.
The {DUSTER} attack: Tor onion service attribution based on flow watermarking with track hiding.
In *22nd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2019)*, pages 213–225, 2019.

References VII

Ioannis Ioannidis, Ananth Grama, and Mikhail J. Atallah.
A secure protocol for computing dot-products in clustered and distributed environments.
In *31st International Conference on Parallel Processing (ICPP 2002), 20-23 August 2002, Vancouver, BC, Canada*, pages 379–384. IEEE Computer Society, 2002.

References VIII

- Rob Jansen, Florian Tschorsch, Aaron Johnson, and Björn Scheuermann.
  The sniper attack: Anonymously deanonymizing and disabling the tor network.
  In *NDSS*, 2014.
- Zhen Ling, Junzhou Luo, Wei Yu, Xinwen Fu, Dong Xuan, and Weijia Jia.
  A new cell-counting-based attack against tor.
  IEEE/ACM Transactions On Networking, 20(4):1245–1261, 2012.

References IX

- Brad Miller, Ling Huang, A.D. Joseph, and J.D. Tygar.
  I know why you went to the clinic: Risks and realization of https traffic analysis.
  http://arxiv.org/abs/1403.0297, 2014.
- David Anthony Stainton.
  Post quantum sphinx.
  Cryptology ePrint Archive, 2023.

Acknowledgements

Co-funded by the European Union (Project 101135475).

Co-funded by SERI (HEU-Projekt 101135475-TALER).

Co-funded by the European Union

Project funded by

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union.

Neither the European Union nor the granting authority can be held responsible for them.