# Genomic Security

(Lest We Forget)

Gene Tsudik

CS@UCI

www.ics.uci.edu/~gts

sprout.ics.uci.edu

# DISCLAIMER

I am:

- A researcher in: security, privacy, applied cryptography

I am **not**:

- An expert in: genomics, genetics, bioinformatics, statistics, ML, and much of everything else

# Genomic Privacy hogs the spotlight!

- Threats appear to be almost immediate, spectacular and terrifying
- Leakage can be direct or indirect, e.g., surname or location inferencing
- Leakage can be massive, e.g., hacked genomic data-banks
- Attack classes:
  - **Large-Scale (impersonal)**: by cyber-criminals, pharmaceuticals, insurance companies, nations
  - **Targeted (personal)**: by competitors, litigants, "friends", relatives, nations
- Progress has been made against large-scale attacks
- But, new ones keep popping up
- Inherent conflict between GWAS needs ("good of the many") and individual privacy needs ("good of the few")

- Also: targeted attacks seem very hard (perhaps impossible) to mitigate

**WHY?**

# We constantly shed DNA material

- Hair (with root)
- Saliva
- Blood
- Skin cells
- Nail clippings (possibly)
- …
- and so on, and so forth

# There is no cure for the focused attack

Not even a full-body condom...

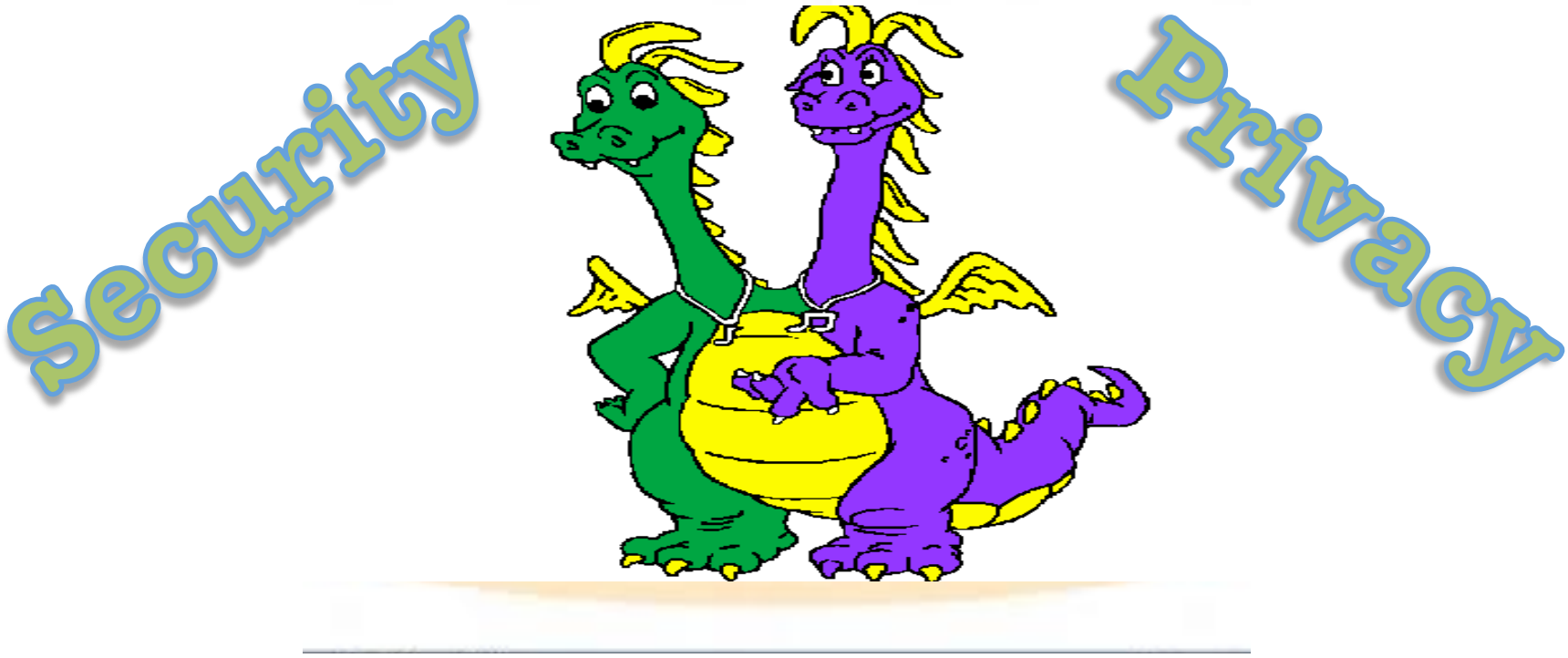
And, let's not forget exhibitionist idiots

# FOR FURTHER INFO, SEE:

https://genomeprivacy.org/

# WHAT ABOUT GENOMIC SECURITY?

## WHY HASN'T IT RECEIVED MUCH ATTENTION?



# Hypothetical Scenario (1)

- Alice gets her genome sequenced by a licensed Sequencing Laboratory (SL)
- Alice's fully sequenced digitized genome is stored on her personal device
- Alice's genome is then modified by:
  - Malware
  - Directly (physically) by adversary
  - Alice herself

- Now what?

# Hypothetical Scenario (2)

- Alice goes to the doctor who treats her condition (e.g., cancer) using personalized medicine. Wrong medicine is administered.
- Alice is admitted to a hospital on emergency basis. Wrong treatment is administered.
- Alice takes part in a parentage test. Wrong outcome!
- Alice submits genomic information to dating app. Gets paired up fraudulently. The horror! ☺

# Security Issues

- Who sequenced the genome?
  - Can that entity be trusted?
  - Who/how certifies this entity?
- Was sequencing done "by the book"?
  - Has the owner consented? or
  - Was the sample otherwise legally obtained?
  - Evidence? Raw data preservation?

- Has the genome been modified?
- Does the genome belong to its claimed owner?
  - How to authenticate the owner?

- Who has the rights/reasons to "see" which portions of the genome?
  - How to authorize, certify, authenticate, etc., such entities?

# Setting, Assumptions, etc.

| | |
|---|---|
| SL | Licensed sequencing laboratory |
| Alice | A human being |
| Tester | Entity authorized to "see" some of Alice's genome<br>• Medical: hospital, clinic, doctor<br>• Legal: court-appointed lab<br>• Social: ancestry or dating app |
| CL | Cloud service provide |
| AUTH | "Higher authority", e.g., FDA |

# Is there **really** a security problem?
# **THERE ISN'T**

**If we abandon privacy**

Security becomes very boring:

- Alice gets signed genome
- Alice gives it to whomever
  - Detail: still need to prove rightful ownership
- That's it…

**Or, if SL and Tester are always one and the same**

**Or, if genomic tests and corresponding regions of the genome are known/fixed**

# A more appealing setting

- Tester and SL are distinct
- Alice and Tester communicate over a network
- Test parameters (positions, ranges) are not pre-fixed

# Requirements (what we want)

- Efficient means for Alice to convince Tester of integrity & authenticity of her (partial) genomic data

- Privacy: reveal to Tester only what's needed, the rest remains secret
  - Ideally, revealed information must not allow Tester to learn anything else (not attainable)

- Performance: minimize storage, communication and computation costs

# Security-Privacy Conflict

- Assume compact (reference) representation
- Each SNP individually signed

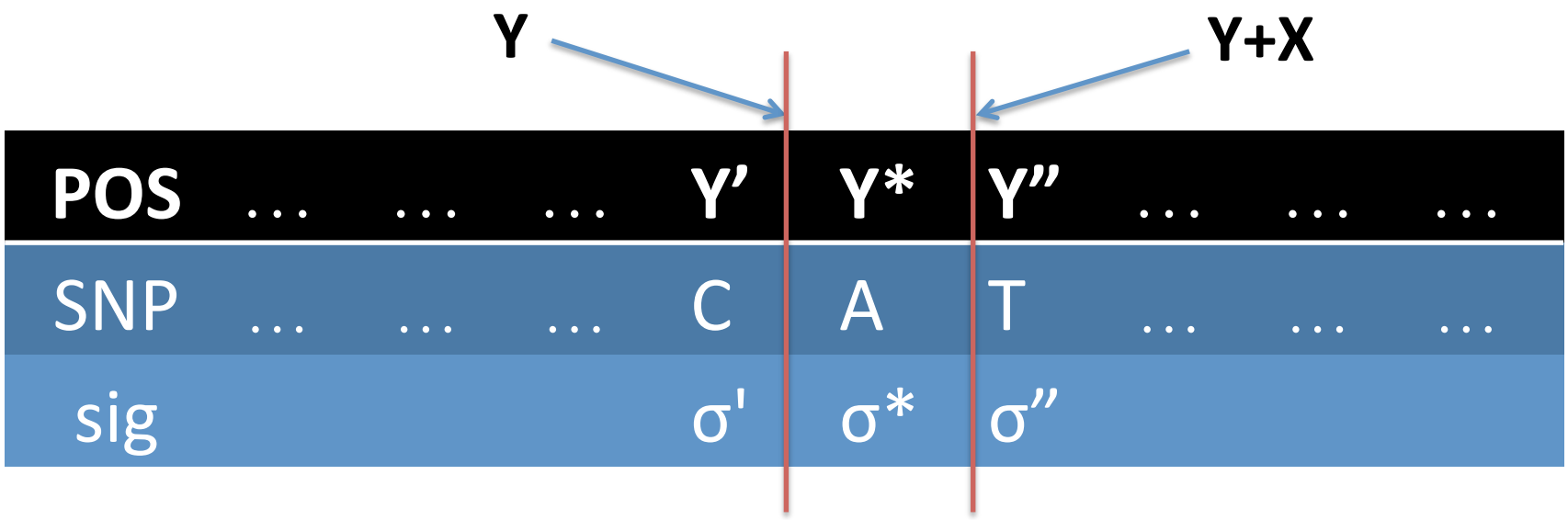
Omission problem:

- Tester asks for mutations in a given range
- Malicious Alice provides some (not all) or claims none
- Can't create new SNPs or modify existing ones, but can omit

Sign ranges instead of individual mutations?

- Not so fast…

# EXAMPLE




- Tester asks for segment of size X, starting at position Y

  $Y>Y'$, $Y<Y^*$, $Y+X<Y''$

- Alice has only one SNP in that range: A at $Y^*$
  - Can provide **$[Y^*, A, \sigma^*]$**, or not…(claim no mutations)
  - How to prove absence of other SNPs in requested range?

Similar to completeness in database range query reply

# EXAMPLE (contd.)

| POS | … | … | … | Y’ | Y* | Y” | … | … | … |
|---|---|---|---|---|---|---|---|---|---|
| SNP | … | … | … | C | A | T | … | … | … |
| SIG | | | | σ' | σ* | σ” | | | |

- Signatures are linked
- No more cheating
- But, Alice would reveal (Y’,σ’) and (Y”,σ”) along with (Y*,σ*)
- Distances: Y-Y’, and Y”-(Y+X) can be VERY LARGE
- Possibly lots of extra information would have to be leaked

- The same holds for other ADS representations, e.g., MHT

# How to avoid leakage?

- Revert to full genome representation…
- Storage is getting cheaper and cheaper
- Alice can store her own entire genome

And then?

- Sign DNA segments (of what size?)

Or:

- Sign each base-letter individually - most flexible

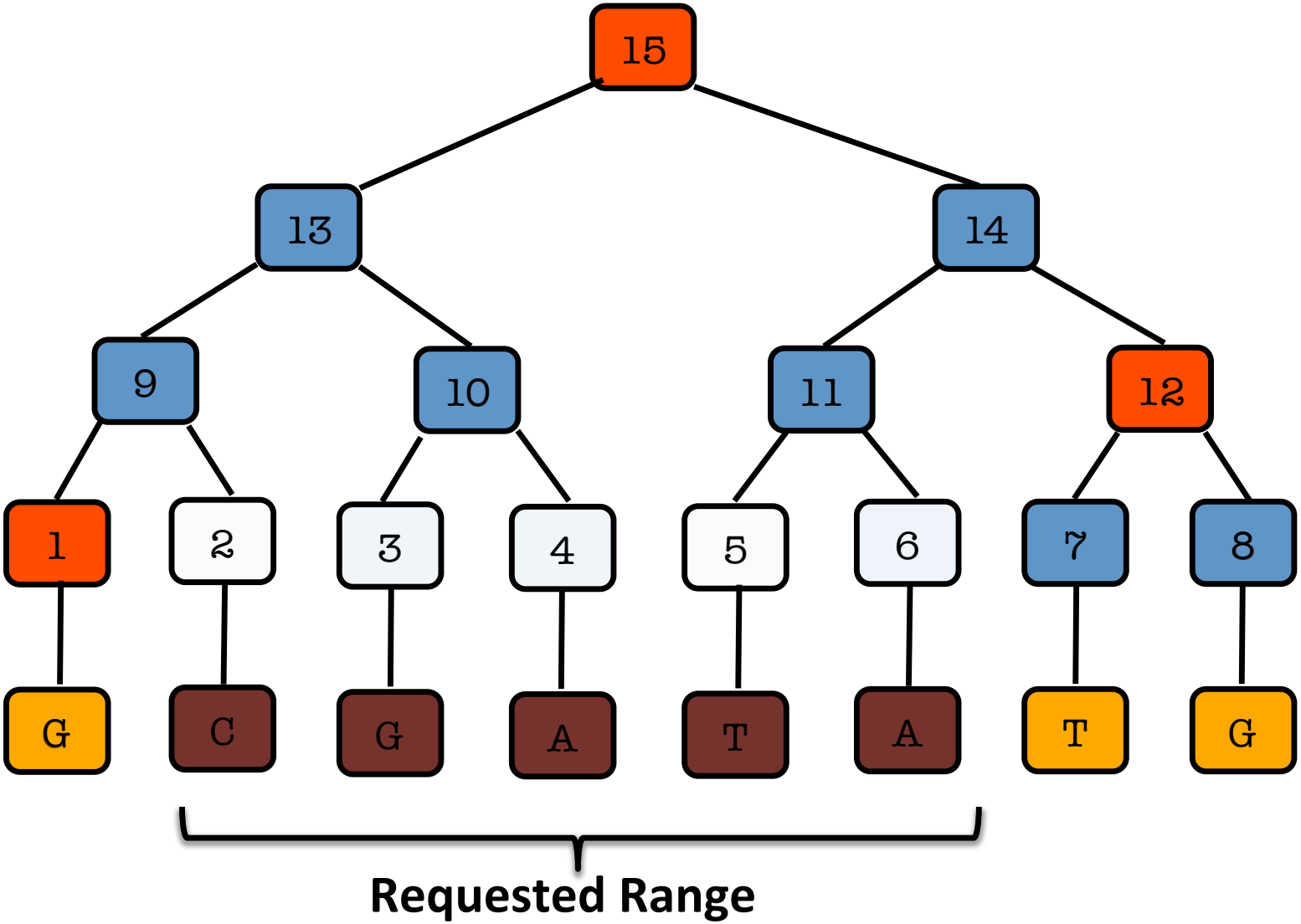
# Overhead…

- Signing → not a problem (SL can do it off-line)
- Extra bits per base-letter: 224 ECC, 2048 RSA
- Transmission and/or verification optimizations:
  - Batch signatures, e.g., w/FDH-RSA, BGR (EC'98)
  - Condensed signatures, e.g., MNT (NDSS'04)
  - Aggregated signatures, e.g., BGLS (EC'03)

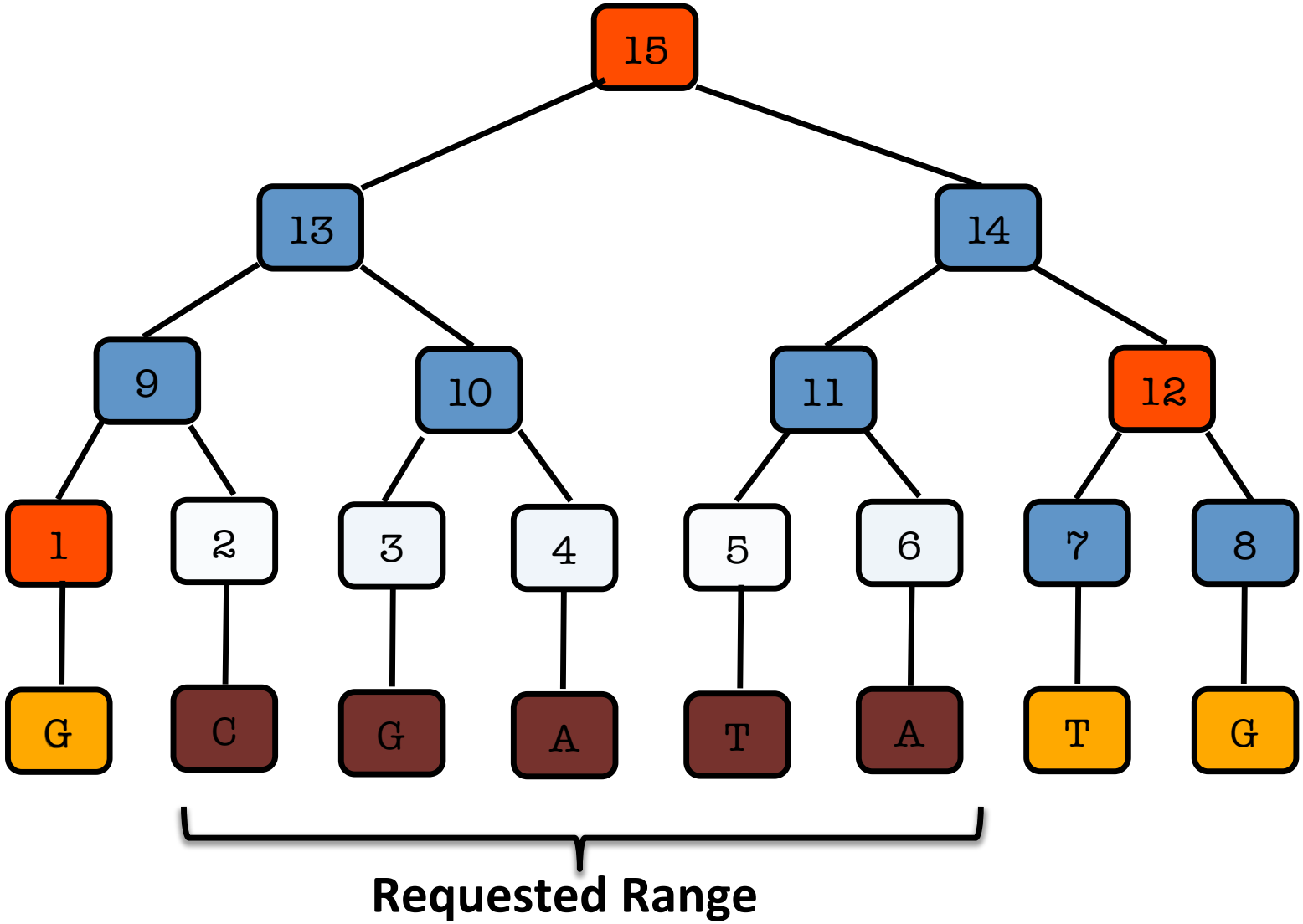
# Merkle Hash Tree

- Security analog of a Phillips screwdriver ☺
- SL builds MHT with base-letters as leaves
- Signs the root
- MHT height ca. 30
- Storage/computation trade-off for Alice
- Low computational costs for Tester
  - About 30 hashes + 1 sig verification
- Could also use other ADS-s, e.g., skip-lists

# Merkle Hash Tree (contd)

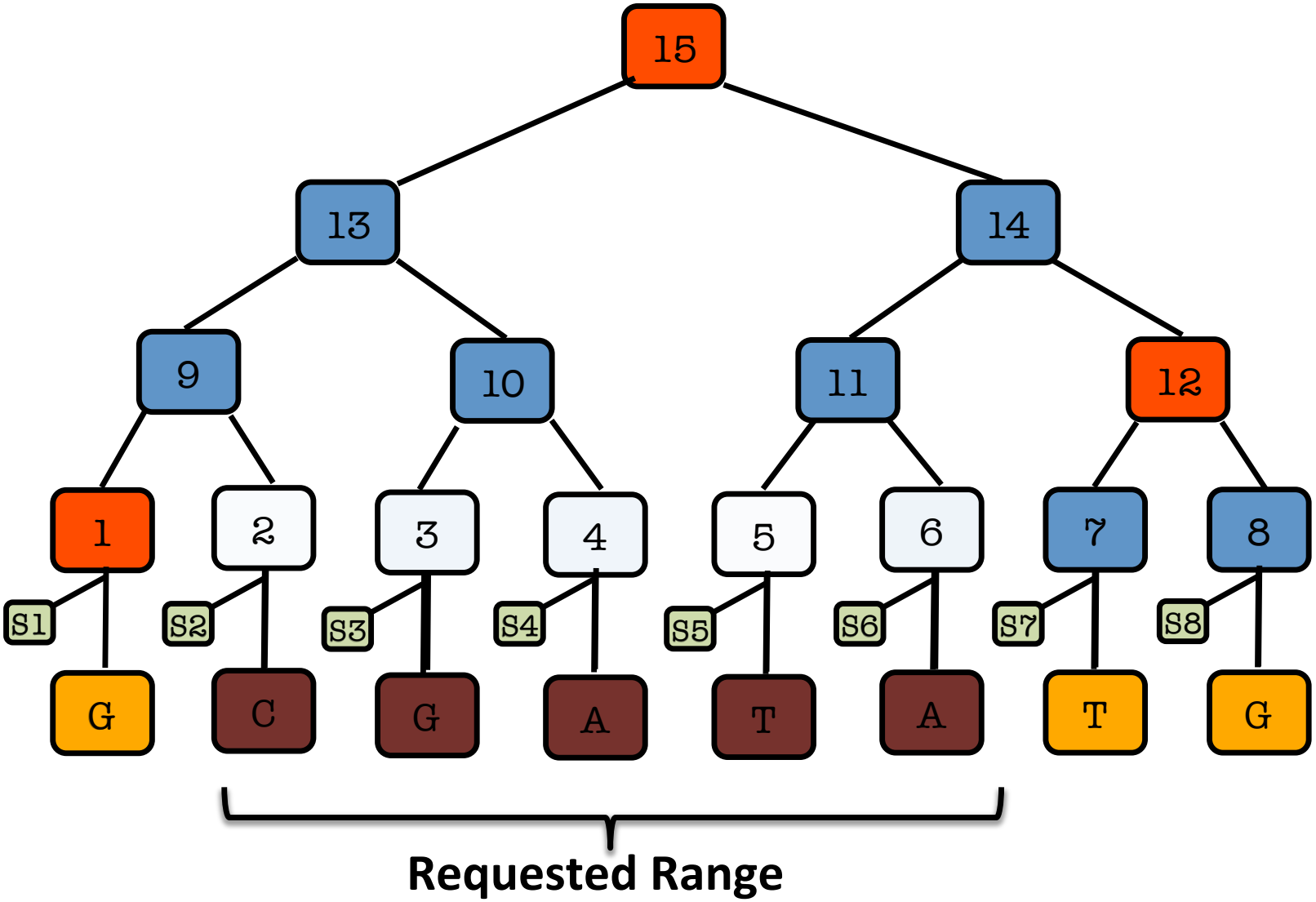


# MHT Leakage Example




- exhaustive search practical up to about height 5, i.e., 32 extra base-letters might be learned by Tester

# How to cure it? Salt the MHT!



# Salted MHT

- Salted by SL at creation time
- Salts generated from master key via PRF
- Key given to Alice
- Salts for requested leaves revealed to Tester

More generally:

- Redactable signatures concept
  - CT-RSA'02, ICISC'01

# A better way: DSAC

- **D**igital **S**ignature **A**ggregation & **C**haining (DSAC)
- Given sequence: $\{L_1,\ldots,L_N\}$, SL computes, for $0<i<N$:

  $(R_0 = s_0)$

$$R_i = [\, L_i,\ i,\ s_i, H(R_{i-1}, s_{i-1}) \,]$$

$$\sigma_i = Fsig\,(R_i)$$

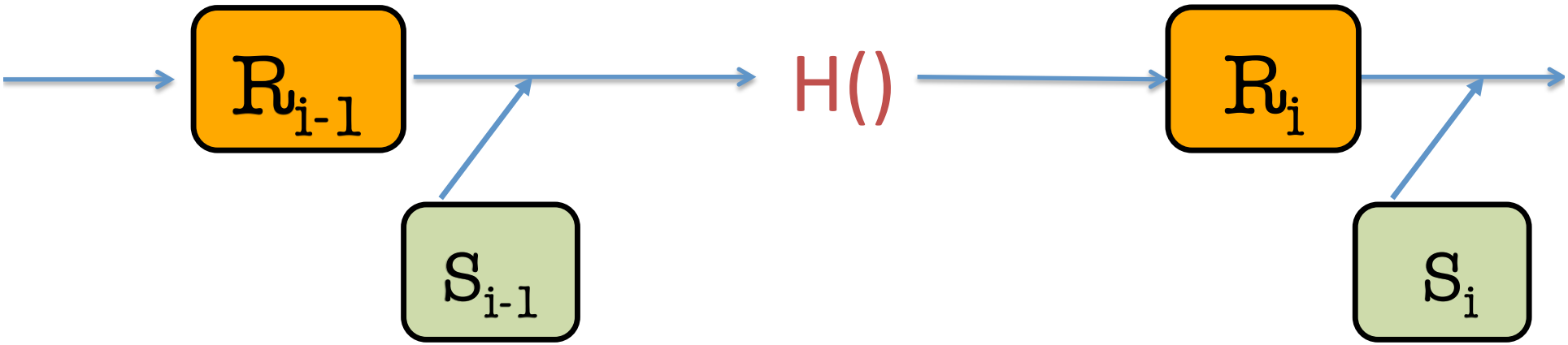
where:

- Fsig() – hash-and-sign signature function
- $s_i,\ldots,s_{iN}$ – pseudo-random salts (like in MHT)
- H() – suitable hash function

# DSAC (contd.)

$$R_i = [\, L_i,\ i,\ s_i, H(R_{i-1}, s_{i-1}) \,]$$

$$\sigma_i = \mathrm{Fsig}\,(R_i)$$



# DSAC (contd.)

- Tester asks for base-letters in range: [ i, j ]
- Alice provides:
  1. { $(L_i, s_i),\ldots,(L_j, s_j)$ }
  2. $H(R_{i-1}, s_{i-1})$
  3. $\sigma_j$

- Low verification cost: 1 signature, (j-i) hashes

- Low communication cost

# Are we done?

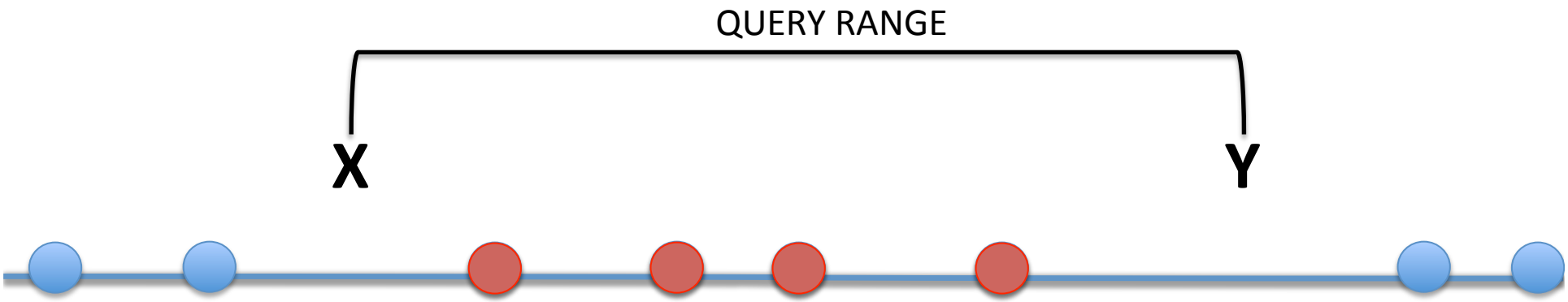
Not yet… only if we're happy with the full representation

**Ideally:**

SL would sign **reference-based** representation, such that Alice can:

- redact arbitrary portions, and
- efficiently prove that ranges requested by Tester are fully represented by combination of: (1) reference genome and (2) non-redacted portions, signed by SL

# PROBLEM: Secure & Private Range Query over Sparse Integers




- **Arbitrary query range**
- **Privacy: no information beyond that in range**
- **Reply completeness: no omissions**
- **Reply authenticity/Integrity: no fake inserts**
- **Efficiency**

# Sketch: Secure & Private Range Query over Sparse Integers

$$R_i = [\ L_i, i, P_i, \ ]$$

$$\sigma_i = \mathrm{Fsig}\ [\ \mathrm{cmt}(P_i), \mathrm{cmt}(R_i), \mathrm{cmt}(R_{i+1}), \mathrm{cmt}(P_{i+1})\ ]$$

LEFT RIGHT

Need: efficient proof of committed exponent in range,
e.g., given a commitment of the form:

$h^A g^B \bmod N$

show that:

B in [V,W]

this is indeed possible, e.g., [Boudot'00], [Chaabouni et al.'09]

# Sketch: Secure & Private Range Query over Sparse Integers

$$R_i = [\, L_i, i, P_i, \,]$$

$$\sigma_i = \mathrm{Fsig}\,[\, \underbrace{\mathrm{cmt}(P_i), \mathrm{cmt}(R_i)}_{\text{LEFT}}, \underbrace{\mathrm{cmt}(R_{i+1}), \mathrm{cmt}(P_{i+1})}_{\text{RIGHT}} \,]$$

① Both within range: open both commitments: LEFT & RIGHT
② LEFT in, RIGHT is not: open LEFT, prove $P_{i+1}$ is outside
③ RIGHT in, LEFT is not, open RIGHT, prove $P_i$ is outside
④ Both out of range (empty range):
- prove $P_{i+1}$ is outside
- prove $P_i$ is outside

# Not quite done

- What if Tester needs to query several ranges (non-contiguous intervals)?
    - Privacy?
- Need progress on redactable signatures and techniques similar to group signature revocation
- ALSO: What if Alice wishes to remain anonymous wrt Tester?

# So…

- Is genomic security under-appreciated?
- Is it important?
- Is it research-worthy?

# For further info, see:

http://ieeexplore.ieee.org/document/8055658/

# THANK YOU

# THE END

QUESTIONS